Energy Efficient AP Selection for Cell-Free Massive MIMO Systems: Deep Reinforcement Learning Approach

Niyousha Ghiasi, Shima Mashhadi, Shahrokh Farahmand, S. Mohammad Razavizadeh, *Senior Member, IEEE*, and Inkyu Lee, *Fellow, IEEE*

***Abstract*—The problem of access point (AP) to device association in a cell-free massive multiple-input multiple-output (MIMO) system is investigated. Utilizing energy efficiency (EE) as our main metric, we determine the optimal association parameters subject to minimum rate constraints for all devices. We incorporate all existing practical concerns in our formulation, including training errors, pilot contamination, and central processing unit access to only statistical channel state information (CSI). This EE maximization problem is highly non-convex and possibly NP-hard. We propose to solve this challenging problem by model-free deep reinforcement learning (DRL) methods. Due to the very large discrete action space of our posed optimization problem, existing DRL approaches can not be directly applied. Thus, we approximate the large discrete action space with either a continuous set or a smaller discrete set, and modify existing DRL methods accordingly. Our novel approximations offer a framework with tolerable complexity and satisfactory performance that can be readily applied to other challenging optimization problems in wireless communication. Simulation results corroborate the superior performance of the modified DRL methods over conventional approaches.**

***Index Terms*—Deep reinforcement learning, Cell-free massive MIMO, Energy efficiency, Pilot contamination, Imperfect CSI.**

I. Introduction

As one of the recommended technologies for 6G implementation [1], cell-free (CF) massive multiple-input multiple-output (MIMO), also known as distributed massive MIMO, enjoys many benefits due to its decentralized nature. Specifically, it can maintain large coverage, provide uniformly good service to all users, and increase diversity due to the favorable propagation conditions. In the CF massive MIMO, every AP can be asked to support all users. This All-AP approach leads to energy inefficiency as distant users suffer from low channel qualities at far-away APs. Subsequently, the additional power consumed by these far-away APs can hardly improve quality of service (QoS) for distant devices. As a result, it is recommended to find a subset of APs for serving each user to optimize energy efficiency. Recently, three general directions have been pursued to address this problem. Heuristic solutions are provided in [2]–[5], while optimization techniques are utilized in [6]–[13]. Also machine learning methods are advocated [14]–[22]. Finally, the performance analysis of AP selection algorithms have been carried out in [23], [24], which utilized stochastic geometry tools to evaluate energy efficiency and delay of an AP selection algorithm.

In [2] and [5], each AP selects N users whose channels have the largest Frobenious norms and then allocates non-zero power control coefficients to the selected users. The goal in [2] is to maximize uplink and downlink energy efficiency (EE), while [4], [5] maximize uplink and downlink spectral efficiency (SE). In [3], a method known as the largest large-scale fading-based selection has been used for the AP selection problem to maximize downlink EE. The proposed solution first selects a subset of APs that form a given percentage of the total channel power for each user and then assigns those APs to that particular user. Power control coefficients are selected afterwards. All these methods are heuristic and do not rely on any optimization formulation. Therefore, if there exist implementation constraints such as the minimum required data-rate for users, it is not possible to incorporate them in this design. In addition, there is a possibility that some users will not be selected by any AP in [2], [5].

AP selection problem is formulated to minimize total consumed power of APs at downlink in [6], [10], while [11] optimizes SE. Maximizing the uplink sum-rate is considered in [12] by utilizing the Hungarian algorithm. However, [12] assumes perfect channel state information (CSI) for the optimization which can be difficult to obtain in practice. Maximization of the worst-case rate of all devices at uplink subject to fronthaul capacity constraints has been investigated in [13], assuming no pilot contamination. Optimizing EE has been investigated by [7]–[9]. In [7], a full-duplex CF massive MIMO system is considered, where a weighted sum of EE and SE is maximized over AP-user associations, uplink power coefficients, sleeping APs, and downlink beamforming vectors. While [7] assumes perfect CSI in its first optimization problem, it formulates a second problem to address pilot contamination by minimizing the worst-case mean-square error (MSE) between different users. Due to separate formulations, imperfect CSI and pilot contamination are not considered in the optimization of weighted sum of EE and SE. In [8], [9], it is proposed to turn a number of APs off in order to maximize EE. The formulated problem in [8] is NP-hard and they resort

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A5A1027646).

N. Ghiasi, S. Mashahdi, S. Farahmand, and S. M. Razavizadeh are with the School of Electrical Engineering, Iran University of Science and Technology (IUST), Tehran, Iran. (emails:{n_ghiasi, sh_mashhadi}@elec.iust.ac.ir,{shahrokhf, smrazavi}@ iust.ac.ir)

I. Lee is with the School of Electrical Engineering, Korea University, Seoul, Korea. (email: inkyu@korea.ac.kr)

S. Farahmand and I. Lee are corresponding authors.

to heuristic methods. In [9], greedy backward search is utilized in order to turn off one AP in each iteration to maximize uplink and downlink energy efficiencies.

Machine learning in general [15]–[17], and deep reinforcement learning (DRL) specifically [18]–[20], [22], have been applied to the AP selection problem in CF massive MIMO. The objective in [20] is to maximize sum-rate, while [19] maximizes minimum rate/signal to interference plus noise ratio (SINR). Maximizing total downlink SE is looked at by [15], [18], while minimum downlink SE is optimized in [16]. In [17], it is assumed that even large-scale fading coefficients are difficult to collect. Hence, it decides AP assignments with partial large-scale fading knowledge using an inductive graph learning framework. Contrary to our EE maximization objective, [17] focuses on finding best graph embeddings. In [22], a general function of signal-to-noise ratios (SNRs) is maximized for downlink in a mmWave setup assuming perfect CSI. The approach selects the optimum analogue beamformers via DRL and optimizes the digital beamformers via convex optimization. However, they do not consider EE. To the best of our knowledge, maximizing EE by selecting AP-user associations via machine learning has only been studied in [21] which assumes perfect instantaneous CSI at the DRL agent. The approach in [21] overlooks all practical considerations such as CSI estimation error, pilot contamination and mere statistical CSI knowledge at the optimizing entity. Given aforementioned references and their limitations, our main contributions can be enumerated as follows:

- We investigate a practical scenario which includes all sources of imperfection such as training error and pilot contamination. Furthermore, we assume that the central processing unit (CPU) knows only statistical CSI. To the best of our knowledge, no other work has considered all these limitations in EE maximization.
- We propose to solve the formulated optimization problem via model-free DRL methods. Given the very large discrete action space of the AP selection problem, available DRL methods cannot be readily applied. Thus, we propose two general modifications to these methods to ensure reduced complexity as well as satisfactory performance. They involve approximating the large discrete action space with either a continuous set or a lower dimensional discrete set.
- Our proposed approximations introduce a framework that can be applied to any optimization problem that demands high complexity. For our EE optimization problem, it is revealed that the proposed algorithms outperform existing approaches.

The rest of this paper is organized as follows: Section II describes the system model and formulates the corresponding optimization problem. Section III provides the two major modifications to available DRL algorithms that will be utilized. Section IV presents numerical results and Section V concludes the paper.

II. System Model and Problem Formulation

We consider a time division duplex (TDD)-based CF massive MIMO system operating in uplink as in Fig. 1. This

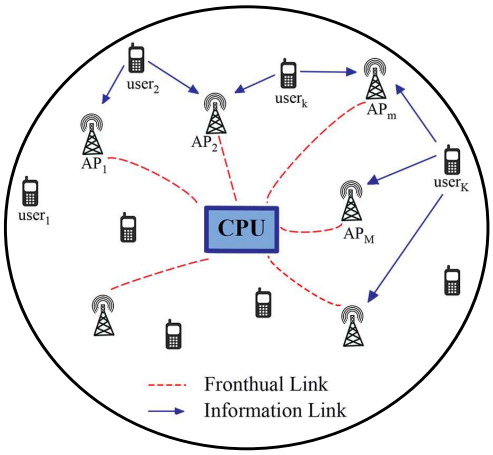

Fig. 1: Cell-free massive MIMO system

system is composed of M APs and K users, which are randomly placed in the environment. Each AP is equipped with O antennas, while each user has a single-antenna. All APs are connected to the central processing unit (CPU) via error-free and high capacity fronthaul links. In TDD-based scenarios, every coherence interval τ_c is divided into three phases: (i) uplink training, (ii) downlink data transmission, and (iii) uplink data communication. Given that we are interested in the uplink mode only, the downlink data transmission phase is not considered. Therefore, by assuming a duration of τ_p for uplink training, the remaining coherence interval, $\tau_c - \tau_p$, is reserved for uplink data communication.

A. Transmission Model

During uplink training, all K users simultaneously broadcast their pilot sequences to all APs, where each AP estimates the channel from all users. Let $\sqrt{\tau_p \zeta_k}\boldsymbol{\phi}_k^H \in \mathbb{C}^{1\times\tau_p}$ denote the transmitted pilot sequence of user k, where $0 < \zeta_k \leq 1$ demonstrates the pilot power control coefficient of the k-th user, and $\boldsymbol{\phi}_{\boldsymbol{k}}^H$ denotes the normalized pilot sequence corresponding to user k with $||\boldsymbol{\phi}_k||^2 = 1$. Then, the m-th AP receives

$$\mathbf{Y}_{p,m} = \sqrt{\tau_p \rho_p} \sum_{k=1}^{K} \mathbf{g}_{mk} \sqrt{\zeta_k} \boldsymbol{\phi}_k^H + \mathbf{W}_{p,m}, \tag{1}$$

where index p indicates the pilot stage, ρ_p equals the SNR of each pilot symbol which is normalized by the received noise power N_0, $\mathbf{W}_{p,m}$ stands for the $O \times \tau_p$ circularly symmetric complex white Gaussian noise matrix whose elements are independent and identically distributed (i.i.d.) $\mathcal{CN}(0,1)$, and $\mathbf{g}_{mk}$ denotes the channel vector between the k-th user and m-th AP given by

$$\mathbf{g}_{mk} = \sqrt{\beta_{mk}} \mathbf{h}_{mk}. \tag{2}$$

Here, β_{mk} is the large-scale fading between the k-th user and m-th AP which dose not depend on the antenna index at the AP, and $\mathbf{h}_{mk}$ represents an $O \times 1$ vector of the small-scale fading. We assume that the elements of $\mathbf{h}_{mk}$ are i.i.d. random variables with $\mathcal{CN}(0,1)$. To estimate the channel gain vector

$\mathbf{g}_{mk}$ for all $k = 1, ..., K$ by the m-th AP, projection of $\mathbf{Y}_{p,m}$ onto $\boldsymbol{\phi}_k$ is carried out

$$\tilde{\mathbf{y}}_{mk} = \mathbf{Y}_{p,m}\boldsymbol{\phi}_k = \sqrt{\tau_p\rho_p}\mathbf{g}_{mk}\sqrt{\zeta_k} + \sqrt{\tau_p\rho_p}\sum_{\ell\neq k}\mathbf{g}_{m\ell}\sqrt{\zeta_\ell}\boldsymbol{\phi}_\ell^H\boldsymbol{\phi}_k + \mathbf{W}_{p,m}\boldsymbol{\phi}_k. \quad (3)$$

The minimum mean-square error (MMSE) estimate of $\hat{\mathbf{g}}_{mk}$ given $\tilde{\mathbf{y}}_{mk}$ is denoted by [25]

$$\hat{\mathbf{g}}_{mk} = \mathbb{E}\{\mathbf{g}_{mk}\tilde{\mathbf{y}}_{mk}^H\}(\mathbb{E}\{\tilde{\mathbf{y}}_{mk}\tilde{\mathbf{y}}_{mk}^H\})^{-1}\tilde{\mathbf{y}}_{mk} = c_{mk}\tilde{\mathbf{y}}_{mk}, \quad (4)$$

where c_{mk} is defined as [26]

$$c_{mk} := \frac{\sqrt{\tau_p\rho_p}\beta_{mk}\sqrt{\zeta_k}}{\tau_p\rho_p\sum_{\ell=1}^{K}\beta_{m\ell}\zeta_\ell|\boldsymbol{\phi}_\ell^H\boldsymbol{\phi}_k|^2 + 1}. \quad (5)$$

The estimated channel $\hat{\mathbf{g}}_{mk}$ includes O components. The mean-square of the o-th component is denoted by γ_{mk}

$$\gamma_{mk} := \mathbb{E}\{|[\hat{\mathbf{g}}_{mk}]_o|^2\} = \sqrt{\tau_p\rho_p}\beta_{mk}\sqrt{\zeta_k}c_{mk}. \quad (6)$$

It should be mentioned that because $\tau_p < K$, pilot sequences for different users are not necessarily orthogonal. Thus, pilot contamination may arise which is modeled in both c_{mk} and γ_{mk}.

Once uplink training is complete, uplink data transmission ensues where all K users simultaneously transmit their data. The received signal at the m-th AP is given by

$$\mathbf{y}_{u,m} = \sqrt{\rho_u}\sum_{k=1}^{K}\sqrt{\eta_k^u}\mathbf{g}_{mk}q_k + \mathbf{w}_{u,m}, \quad (7)$$

where ρ_u is the maximum uplink SNR of user k normalized by noise power N_0, $0 < \eta_k \leq 1$ denotes the data power control coefficient of the k-th user, q_k represents the symbol of the k-th user with $\mathbb{E}[|q_k|^2] = 1$, and $\mathbf{w}_{u,m}$ indicates the noise vector with $\mathcal{CN}(0, 1)$ components.

Given that some APs are far-away from a specific user, it is not efficient to utilize all APs for combining the signals of that specific user. Therefore, we allow each AP to serve only a subset of users. As a result, we define the assignment variables $\alpha_{mk} \in \{0, 1\}$ for $k = 1, ..., K$, and $m = 1, ..., M$, which equals 1 when the k-th user is served by the m-th AP, otherwise, $\alpha_{mk} = 0$. Also, we assume maximum ratio combining (MRC) for all users. Subsequently, $\sum_{o=1}^{O}\alpha_{mk}[\hat{\mathbf{g}}_{mk}]_o^*[\mathbf{y}_{u,m}]_o$ is transmitted to the CPU via the fronthaul links, and the CPU detects q_k from the received signal $r_{u,k}$ given as

$$r_{u,k} = \sum_{m=1}^{M}\sum_{o=1}^{O}\alpha_{mk}[\hat{\mathbf{g}}_{mk}]_o^*[\mathbf{y}_{u,m}]_o. \quad (8)$$

Assuming that the CPU only knows the channel statistics, the received signal $r_{u,k}$ can be rewritten as [27]

$$r_{u,k} = d_k\ q_k + b_k\ q_k + \sum_{\ell\neq k}u_{k\ell}\ q_\ell + z, \quad (9)$$

where d_k, b_k, $u_{k\ell}$, and z are respectively the strength of the desired signal, the beamforming gain uncertainty, the interference caused by the ℓ-th user, and the received effective noise, and are given by

$$d_k = \sqrt{\rho_u}\mathbb{E}\left\{\sum_{m=1}^{M}\sum_{o=1}^{O}\alpha_{mk}\sqrt{\eta_k^u}[\mathbf{g}_{mk}]_o[\hat{\mathbf{g}}_{mk}^*]_o\right\}, \quad (10)$$

$$b_k = \sqrt{\rho_u}\left(\sum_{m=1}^{M}\sum_{o=1}^{O}\alpha_{mk}\sqrt{\eta_k^u}[\mathbf{g}_{mk}]_o[\hat{\mathbf{g}}_{mk}^*]_o - \mathbb{E}\left\{\sum_{m=1}^{M}\sum_{o=1}^{O}\alpha_{mk}\sqrt{\eta_k^u}[\mathbf{g}_{mk}]_o[\hat{\mathbf{g}}_{mk}^*]_o\right\}\right), \quad (11)$$

$$u_{k\ell} = \sqrt{\rho_u}\sum_{m=1}^{M}\sum_{o=1}^{O}\alpha_{mk}\sqrt{\eta_\ell^u}[\mathbf{g}_{m\ell}]_o[\hat{\mathbf{g}}_{mk}^*]_o, \quad (12)$$

$$z = \sum_{m=1}^{M}\sum_{o=1}^{O}\alpha_{mk}[\mathbf{w}_{u,m}]_o\hat{[\mathbf{g}}_{mk}^*]_o. \quad (13)$$

Using the worst-case Gaussian noise argument [28], the spectral efficiency of the k-th user is computed by

$$R_k = \frac{\tau_c - \tau_p}{\tau_c}\times \log_2\left(1 + \frac{|d_k|^2}{\mathbb{E}\{|b_k|^2\} + \sum_{\ell\neq k}^{K}\mathbb{E}\{|u_{k\ell}|^2\} + \mathbb{E}\{|z|^2\}}\right). \quad (14)$$

Upon evaluating the expected values in (10), (11), (12), (13), a closed form expression for the uplink spectral efficiency of the k-th user is obtained as (15) on top of next page [26] where

$$\chi := \sum_{m=1}^{M}\alpha_{mk}\gamma_{mk}\sqrt{\eta_\ell^u}\sqrt{\frac{\zeta_\ell}{\zeta_k}}\frac{\beta_{m\ell}}{\beta_{mk}}. \quad (16)$$

The sum of the spectral efficiency of users is given by

$$\mathrm{SE}_u = \sum_{k=1}^{K}R_k. \quad (17)$$

B. Power Consumption Model and Energy Efficiency

The total uplink power consumption is modeled as [10]

$$P_{total} = \rho_u N_0\sum_{k=1}^{K}\frac{1}{\nu_k}\eta_k^u + \sum_{m=1}^{M}P_m + \sum_{m=1}^{M}P_{fh,m}. \quad (18)$$

The first term in (18) represents the power consumption of users for transmitting data, where $0 < \nu_k \leq 1$ indicates the power amplifier efficiency at the k-th user, and N_0 is the received noise power in the receiver. The second term equals the power consumption for RF chains, antennas, and processing circuit of all APs, and the third term is related to the power consumption of the fronthaul network. It is assumed that processing circuit power is significantly smaller than the RF chains. Then, the antenna power consumption becomes

$$P_m = \mathrm{sgn}\Big(\sum_{k=1}^{K}\alpha_{mk}\Big)OP_{tc,m}, \quad (19)$$

where $\mathrm{sgn}(.)$ represents the sign function, $P_{tc,m}$ is the internal power for each antenna of the m-th AP, required to run the circuit components (e.g. converters, mixers, and filters). According to (19), if the m-th AP does not serve any user, the RF chains and antenna power consumption P_m will be zero. The fronthaul network power consumption $P_{fh,m}$ is given as

$$P_{fh,m} = P_{fix,m} + B\sum_{k=1}^{K}\alpha_{mk}R_kP_{ft,m}. \quad (20)$$

$$R_k = \frac{\tau_c - \tau_p}{\tau_c} \log_2 \left(1 + \frac{O^2 \rho_u \big(\sum_{m=1}^{M} \alpha_{mk} \sqrt{\eta_k^u} \gamma_{mk} \big)^2}{O^2 \rho_u \sum_{\ell \neq k} (\chi)^2 |\boldsymbol{\phi}_\ell^H \boldsymbol{\phi}_k|^2 + O \rho_u \sum_{\ell=1}^{K} \sum_{m=1}^{M} \alpha_{mk} \eta_\ell^u \gamma_{mk} \beta_{m\ell} + O \sum_{m=1}^{M} \alpha_{mk} \gamma_{mk}} \right), \tag{15}$$

The first term in (20) represents the fixed power consumption of each fronthaul link, which depends on the system topology and the distances between the APs and the CPU. In contrast, the second term in (20) $P_{ft,m}$ depends on data traffic (in Watt per bit/s), because the fronthaul network is used to transfer the data between the APs and the CPU, so its power consumption is proportional to the sum of spectral efficiency. The total uplink energy efficiency (bit/Joule) is defined as the sum of the uplink spectral efficiency of users (bit/s) divided by the total uplink power consumption (Watt) in the network, as

$$\mathrm{EE} = \frac{B \times \mathrm{SE}_u}{P_{total}}, \tag{21}$$

where B is the system bandwidth.

C. Problem Formulation

In this paper, we address the problem of selecting APs to serve each user in order to maximize the energy efficiency under the constraints on per-user spectral efficiency as

$$\max_{\boldsymbol{\alpha}=\{\alpha_{k,m}\}} \quad \mathrm{EE} \tag{22a}$$

$$\text{s.t.} \quad \alpha_{mk} \in \{0,1\}, \quad \forall k, m \tag{22b}$$

$$R_k \geq R_{th}, \quad \forall k \tag{22c}$$

where R_{th} is the minimum spectral efficiency required for the k-th user. We assume the same minimum R_{th} for all users because CF systems have the advantage of providing satisfactory uniform service to all network users. Problem (22) is solved by the CPU in a centralized fashion. It is noteworthy that our objective contains all practical sources of imperfection that can arise in a CF setup. We have modeled pilot contamination as well as imperfect and noisy CSI at APs. Also, we assume only statistical CSI knowledge at the CPU thus obviating the need to communicate small-scale fading coefficients from APs to CPU. The problem in (22) is an integer program and the feasible region consists of 2^{MK} discrete points without constraint (22c). If we consider (22c), some of the discrete points might become infeasible. Due to its discrete structure, the feasible region is non-convex and the formulated problem is non-differentiable. Furthermore, both EE and rates are highly non-concave functions of α_{mk}'s. To solve this problem efficiently, we propose DRL methods that can achieve sub-optimal solutions yielding satisfactory objective values in the following section.

III. Proposed DRL-Based AP Selection

Given the plethora of existing DRL methods, we limit our attention to model-free algorithms that are well-suited to solve optimization problems. Being model-free, these algorithms do not rely heavily on the posed problem structure, thus they can be applied to any optimization problem with minor modifications. In Sub-Section III.A, RL concepts are briefly introduced and their corresponding parameters are specified for our problem. Then, we motivate why DRL is needed instead of RL to solve (22). In Sub-Sections III.B and III.C, we present two general directions to modify existing DRL algorithms to tailor them for efficiently solving (22).

A. Reinforcement Learning (RL)

To describe RL, the following definitions for five important elements are needed [29].

- State space S is the set of states s observed by an agent in the environment. In our problem, the large scale fading coefficients of channels β_{mk} are considered as the state space. Thus, the state space maintains KM positive continuous variables.
- Action space A is the set of actions a taken by an agent in each state. In our AP selection (APS) problem, joint selection of all α_{mk} form the action space composed of 2^{KM} discrete points.
- Immediate reward function $r(s,a)$ for taking action $a \in A$ in state $s \in S$ is considered as

$$r(s,a) = \mathrm{EE} - \lambda \sum_{k=1}^{K} u_0(R_{th} - R_k), \tag{23}$$

  where $u_0(x)$ is a step function which is zero for $x \leq 0$ and one, otherwise. It should be pointed out that if we penalize the SE constraints in (22c) and bring them into the objective, we can form the Lagrangian. Unfortunately the problem (22) is non-convex and hence strong duality does not hold. It is notable that step functions are added in (23) to create a uniform penalty for all users whose minimum rates are not met. As λ grows large, solving (23) will be equivalent to solving (22).
- Policy π which could be stochastic i.e., a distribution over actions $\pi(a|s)$ given we are in state s, or deterministic $\pi(s) \in A$. The stochastic policy determines the probability of taking each action $a \in A$ according to the observed state $s \in S$, and the deterministic policy maps the observed state $s \in S$ to the actions $a \in A$ that will be taken by an agent in those states.
- State-action-value function $q_\pi(s,a)$ which is the long-term reward that is defined as the expected cumulative discounted reward in the future for the action $a \in A$ that is taken by an agent in the state $s \in S$ under policy π.

RL aims to select the optimal policy $\pi^*(s)$ for every state s that maximizes $q_\pi(s,a)$. Upon defining a discount factor $\eta \in [0,1]$, $q_\pi(s,a)$ is expressed as

$$q_\pi(s,a) = E\left\{ \sum_{t=0}^{\infty} \eta^t r(s_t, a_t) | s_0 = s, a_0 = a \right\}, \tag{24}$$

where expected value is taken over Markov transition probabilities $p_{s_t s_{t+1}}(a_t)$. The optimal policy per state is found by

$$\pi^*(s) = \arg \max_{a \in A} q_\pi(s, a). \tag{25}$$

When both S and A are discrete and finite, Q-learning algorithm [29] can be applied to solve (25) [30]. However, in our problem S is continuous as large-scale fading values can assume any positive value. Furthermore, A is finite but extremely large. Subsequently, Q-learning can not be directly applied. Thus, we rely on deep neural networks (DNNs) to implement reinforcement learning.

Most DRL algorithms are designed to handle continuous states as in deep Q-learning (DQL) [30]. However, a large discrete action space poses a major challenge for them. Subsequently, we propose two general directions to modify existing DRL methods and tailor them to handle large discrete action spaces. First approach is to approximate the large discrete structure with a continuous set. After obtaining the best continuous action, we round it to the nearest available discrete action. Two well-known algorithms designed with continuous action profiles in mind are continuous soft actor critic (C-SAC) [31] and deep deterministic policy gradients (DDPG) [32]. While C-SAC finds optimum stochastic policies, DDPG finds optimum deterministic policies. We will present both DDPG and C-SAC modifications in Sub-section III.B.

Second direction for DRL modification is to utilize a method designed for discrete action space, but introduce certain approximations in their DNN structure to ensure a satisfactorily small action space at the output layer of DNN. Two well-know algorithms designed for discrete action profiles are discrete soft actor critic (D-SAC) [33] and deep Q-learning (DQL) [30]. While D-SAC finds optimum stochastic policies, DQL finds optimum deterministic policies. We will present D-SAC modifications in Sub-section III.C. However, we will skip DQL as its performance was not satisfactory in the simulations. It needs to be noted that complete derivations of these methods are out of the scope of this work. Thus, we only provide algorithmic details of each and direct the reader to corresponding references for the complete derivation.

Algorithm 1: Modified DDPG Algorithm

Randomly initialize CN $Q_\theta(s, a)$ and AN $\pi_\phi(s)$ with weights θ and ϕ.
Initialize target networks $Q_{\bar{\theta}}(s, a)$ and $\pi_{\bar{\phi}}(s)$ with $\bar{\theta} \leftarrow \theta, \bar{\phi} \leftarrow \phi$.
Initialize replay buffer D which is empty.
Fix a given value for the covariance matrix Υ, initial state s_1, and target update rate $\kappa \in (0, 1]$.
for $t \geq 1$ **do**
  Generate a random vector $V_t \sim \mathcal{N}(0, \Upsilon)$.
  Select continuous action vector $a_t = \pi_\phi(s_t) + V_t$.
  Set each element of action vector $\boldsymbol{\alpha}$ to one ($\alpha_{mk} = 1$) if $[a_t]_{mk}$ is greater than zero, otherwise set $\alpha_{mk} = 0$.
  Execute action $\boldsymbol{\alpha}$ and observe reward $r(s_t, a_t)$ and new state s_{t+1}.
  Store transition $(s_t, a_t, r(s_t, a_t), s_{t+1})$ in D.
  Sample a random batch of maximum $\tilde{N}$ transitions from $\{(s_t^i, a_t^i, r(s_t^i, a_t^i), s_{t+1}^i)\}_{i \in D}$ which we represent by D_t.
  Set $Q_{tar}(s_t^i, a_t^i) = r(s_t^i, a_t^i) + \eta Q_{\bar{\theta}}(s_{t+1}^i, \pi_{\bar{\phi}}(s_{t+1}^i))$ for all $i \in D_t$.
  Update CN weights θ by implementing one steepest descent step to minimize the loss function: $J(\theta) = \frac{1}{|D_t|} \sum_{i \in D_t} (Q_{tar}(s_t^i, a_t^i) - Q_\theta(s_t^i, a_t^i))^2$.
  Update AN weights ϕ by one step of steepest ascent on the following objective function: $J_\pi(\phi) = \frac{1}{|D_t|} \sum_{i \in D_t} Q_\theta\left(s_t^i, \pi_\phi(s_t^i)\right)$.
  Update the target networks weights as

$$\bar{\theta} \leftarrow \kappa\theta + (1 - \kappa)\bar{\theta}, \tag{26}$$

$$\bar{\phi} \leftarrow \kappa\phi + (1 - \kappa)\bar{\phi}. \tag{27}$$

  $t \leftarrow t + 1$
end

B. Continuous Approximation of Discrete Action Space

When dealing with continuous states and actions, the most prevalent approach is to utilize separate actor and critic networks, where actor network (AN) decides the next action and critic network (CN) predicts the Q-value for that specific action. To elaborate, current state is fed to the actor as input, where an action is produced as output. Then, the proposed action alongside the current state are fed to the critic, which yields the predicted Q-value for the given state/action pair. Afterwards, the action is taken and the immediate reward obtained is stored in replay memory. These information are later utilized to update AN/CN weights. This type of algorithms are generally referred to as actor-critic (AC). Both DDPG and C-SAC utilize this structure but with certain differences.

In order to solve problem (22) with DDPG, we approximate every α_{mk} with a continuous value. The states and rewards are similar to those discussed in the previous subsection. To ensure better stability of the algorithm, it is common practice to define two DNNs for actor and critic. The one utilized for decision making is usually referred to as target network, while the other network weights are updated by the algorithm. Every once in a while target network weights are substituted by the other trained network weights. Proceeding with DDPG formulation, target actor network (TAN) receives KM large-scale fading coefficients of channel as input and outputs KM continuous actions corresponding to α_{mk}s. Since, we apply a hyperbolic tangent function in going from the last hidden layer of AN to the output layer, all actions ranges are in the interval $[-1, 1]$. Target critic network (TCN) gets a pair (s, a) with $2KM$ dimensions as input and determines $Q_{\bar{\theta}}(s, a)$, where $\bar{\theta}$ represents the TCN weights. After taking the action and observing the reward, AN and CN weights are updated. Lastly, to determine the values of each α_{mk}, if the continuous action proposed by TAN is greater than zero, we set $\alpha_{mk} = 1$, otherwise we set $\alpha_{mk} = 0$. Our modified DDPG is summarized in Algorithm 1.

As a common approach to ensure proper exploration in AC methods, an entropy term is added to the objective function leading to soft AC (SAC). Given that the additive term corresponds to the entropy of policy, it will enforce some randomness, and thus exploration, in the policy. The optimal stochastic policy π^* is obtained from maximizing the following objective function [31]

$$\pi^* = \arg \max_{\pi} \sum_{t} \mathbb{E}_{(s_t,a_t)\sim p}\Big[\eta^t\Big(r(s_t,a_t)+\psi H\big(\pi(a_t|s_t)\big)\Big)\Big], \tag{28}$$

where p is the (unknown) Markov transition probabilities, $H\big(\pi(a_t|s_t)\big)$ is equal to $-\log\big(\pi(a_t|s_t)\big)$, and ψ is the weight that determines the importance of the entropy value relative to the reward, and thus controls the randomness of the optimal policy. The advantage of using this objective function is the wider exploration of the environment, which in turn increases the learning speed of the algorithm. One limitation of this objective function corresponds to the optimal method to tune the weight as it is difficult to tune it manually. A systematic approach to tune ψ can be found in [31] and we utilize it here as well.

To solve problem (22) with C-SAC, the states and rewards are fixed as in previous sections, and each α_{mk} is treated as a continuous action. AN requires MK input neurons and $2MK$ output neurons to get the large scale fading coefficients β_{mk} as inputs and provide the mean and variance of each α_{mk} as output. Moreover, CN and TCN require $2MK$ input neurons and one output neuron to receive the large scale fading coefficients β_{mk} and the selected actions provided by ANs as inputs and output the state-action-value of the observed state and the selected action. To select actions, a Gaussian distribution is constructed for each α_{mk} with the mean and variance obtained from AN. In the exploration phase, each distribution is sampled and in the exploitation phase, the mean of each α_{mk} obtained from AN is used as the selected sample. Then tanh is applied to the selected samples to bring them in the limited range of [-1,1]. Subsequently, if the obtained sample is greater than zero, $\alpha_{mk} = 1$, otherwise $\alpha_{mk} = 0$. It should be noted that C-SAC does not have a TAN, but utilizes two CNs and two TCNs to mitigate positive bias [31]. Modified C-SAC is summarized in Algorithm 2.

C. Low-Dimension Discrete Approximation of Large Discrete Action Space

D-SAC is similar to C-SAC but it is designed for discrete action spaces. The only difference between C-SAC and D-SAC is that the output of AN for D-SAC maintains as many neurons as there are actions and the value of each neuron directly determines the value of $\pi_\phi(a_t|s_t)$ for that particular action instead of specifying the mean and variance of the policy. In D-SAC, CN only receives the state as input, and provide as output the state-action-values for all possible actions given the observed state [33]. To implement D-SAC, the state and reward are similar to those discussed in the previous subsections. Since the AP selection coefficients α_{mk} form the action space, and they assume binary values, the action space is of size 2^{MK} which is too high to implement in a DNN.

Algorithm 2: Modified C-SAC Algorithm

1 Randomly initialize AN weights ϕ and CNs weights θ_1 and θ_2.
2 Initialize TCNs with $\bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2$.
3 Initialize replay buffer D which is empty, and the initial state s_1.
4 **for** $t \geq 1$ **do**
5 Sample vector ϵ_t from standard jointly Gaussian distribution.
6 Set $[a_t]_{mk} = [\mu_\phi(s_t)]_{mk} + [\epsilon_t]_{mk}[\sigma_\phi(s_t)]_{mk}$ for all m, k, where μ_ϕ, σ_ϕ's are AN outputs from input s_t.
7 Apply tanh to each element of a_t to project them to the limited range $[-1, 1]$.
8 Consider action $\alpha_{mk} = 1$ if $[a_t]_{mk}$ is greater than zero, otherwise set $\alpha_{mk} = 0$.
9 Execute action $\boldsymbol{\alpha}$ in the environment. Observe next state s_{t+1} and reward $r(s_t, a_t)$
10 Store $(s_t, a_t, r(s_t, a_t), s_{t+1})$ in replay buffer D.
11 Sample a random batch of maximum $\tilde{N}$ transitions from $\{(s_t^i, a_t^i, r(s_t^i, a_t^i), s_{t+1}^i)\}_{i\in D}$ which we represent by D_t.
12 For all $i \in D_t$, generate a_{t+1}^i according to steps 5-7 but with s_t replaced with s_{t+1}^i.
13 Evaluate $Q_{\min}^i := \min\big\{Q_{\bar{\theta}_1}(s_{t+1}^i, a_{t+1}^i), Q_{\bar{\theta}_2}(s_{t+1}^i, a_{t+1}^i)\big\}$.
14 Update CNs by a single steepest descent step on the objective as

$$J_Q(\theta_1) = \frac{1}{|D_t|}\sum_{i\in D_t}\Big[\big(Q_{\theta_1}(s_t^i, a_t^i) - \big[r(s_t^i, a_t^i) + \eta\left(Q_{\min}^i - \psi\log(\pi_\phi(a_{t+1}^i|s_{t+1}^i))\right)\big]\big)^2\Big],$$

where same computation should be carried out for θ_2 as well.
15 For all $i \in D_t$, generate a vector $\tilde{\epsilon}_t^i$ whose entries come from a standard Gaussian distribution.
16 Update AN by a single steepest ascent step on the objective as

$$J_\pi(\phi) = \frac{1}{|D_t|}\sum_{i\in D_t}\Big[Q_\theta\left(s_t^i, \mu_\phi(s_t^i) + \tilde{\epsilon}_t^i\sigma_\phi(s_t^i)\right) - \psi\log\left(\pi_\phi\left(\mu_\phi(s_t^i) + \tilde{\epsilon}_t^i\sigma_\phi(s_t^i)\right)\right)\Big],$$

where the value of Q_θs appearing in the gradient is substituted by the minimum of two CNs.
17 Update TCNs by the weights of CNs after T_{C-SAC} steps.
18 $t \leftarrow t + 1$
19 **end**

Thus, we propose the following modification to actor and critic networks compared to the original D-SAC algorithm. We assume that each AP-user association is selected independently from others. Thus, we have KM actions that can be either zero or one. Subsequently, every output neuron represents the

Q-function for $Q(s, a_{mkj})$, for $m = 1, \ldots, M$, $k = 1, ..., K$, $j = 0, 1$, and j represents if the corresponding association was made or not (one or zero). Thus our DNN for Q will have $2KM$ output neurons and KM input neurons for the large-scale fading coefficients. For each m, k, in the exploration phase $Q(s, a_{mkj})$ is selected randomly from $\{0, 1\}$ and in the exploitation phase the highest Q value $Q(s, a_{mkj})$ between $j = 0, 1$ is selected for every a_{mk}. Thus, Q-values for every a_{mk} is evaluated separately. In order to minimize the loss function for training CNs, the same reward is assigned to all the assignments independently. Thus, average value of the Q values and the target Q's over KM selections are obtained and used for estimating loss function, which is given by

$$J_Q(\theta) = \mathbb{E}_{s_t \sim D}\Big[\frac{1}{KM}\sum_{m=1}^{M}\sum_{k=1}^{K}\Big(\frac{1}{2}\Big(Q_\theta(s_t, a_{tmkj}) - \Big(r(s_t, a_{tmkj}) + \eta \sum_{j=0}^{1}\pi(a_{(t+1)mkj}|s_{t+1}) \times \big(Q_{\bar{\theta}}(s_{t+1}, a_{(t+1)mkj}) - \psi \log(\pi_\phi(a_{(t+1)mkj}|s_{t+1}))\big)\Big)^2\Big)\Big],$$

where D denotes the replay memory of all previous state-action-reward-next state experiments. Furthermore, $r(s_t, a_{tmkj})$ is the same for all selected actions a_{tmkj} at time step t and obtained from (23). The loss function that must be minimized for updating the weight vector ϕ for the AN is defined as

$$J_\pi(\phi) = \mathbb{E}_{s_t \sim D}\Big[\frac{1}{KM}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=0}^{1}\pi(a_{tmkj}|s_t) \times \big[\psi \log(\pi(a_{tmkj}|s_t)) - Q_\theta(a_{tmkj}, s_t)\big]\Big].$$

Focusing on AN, for each a_{mk}, there exist two output neurons in AN which indicate the probability that a_{mk} is zero or one via a soft-max function. As a result, AN requires MK input neurons and $2MK$ output neurons to receive the large scale fading coefficients β_{mk} as inputs and provide $\pi(a_{tmkj}|s_t)$ for $m = 1, ..., M$, $k = 1, \ldots, K$, and $j = 0, 1$ as output. For each m, k, in the exploration phase, the action a_{tmkj} is selected by sampling from the exact action distribution $\pi(a_{tmkj}|s_t)$ provided by the AN. In the exploitation phase, the action with the highest probability is selected. A summary of D-SAC is provided in Algorithm 3.

Algorithm 3: Modified D-SAC Algorithm

1. Randomly initialize AN weights ϕ and CNs weights θ_1 and θ_2.
2. Initialize TCNs with $\bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2$.
3. Initialize replay buffer D which is empty, and the initial state s_1.
4. **for** $t \geq 1$ **do**
5. Select action $[a_t]_{mkj} \sim \pi_\phi([a_t]_{mkj}|s_t)$ for all m, k and $j = 0, 1$.
6. Execute action a_t in the environment.
7. Observe next state s_{t+1} and reward $r(s_t, a_t)$. Store $(s_t, a_t, r(s_t, a_t), s_{t+1})$ in D.
8. Sample a random batch of maximum $\tilde{N}$ transitions from $\{(s_t^i, a_t^i, r(s_t^i, a_t^i), s_{t+1}^i)\}_{i \in D}$ which we represent by D_t.
9. For all $i \in D_t$, generate a_{t+1}^i according to steps 5 but with s_t replaced with s_{t+1}^i.
10. Evaluate $Q_{\min}^i := \min\big\{Q_{\bar{\theta}_1}(s_{t+1}^i, a_{t+1}^i), Q_{\bar{\theta}_2}(s_{t+1}^i, a_{t+1}^i)\big\}$.
11. Update CNs by a single steepest descent step on the objective:

$$J_Q(\theta_1) = \frac{1}{|D_t|}\sum_{i \in D_t}\frac{1}{KM}\sum_{k=1}^{K}\sum_{m=1}^{M}\Bigg[\Bigg(Q_{\theta_1}(s_t^i, [a_t^i]_{mkj}) - \Big[r(s_t^i, a_t^i) + \eta\sum_{j=0}^{1}\pi([a_{t+1}^i]_{mkj}|s_{t+1}^i)\big(Q_{\min}^i - \psi\log\big(\pi_\phi([a_{t+1}^i]_{mkj}|s_{t+1})\big)\big)\Big]\Bigg)^2\Bigg],$$

12. where same computation should be carried out for θ_2 as well.
13. Update AN by a single steepest ascent step on the objective:

$$J_\pi(\phi) = \frac{1}{|D_t|}\sum_{i \in D_t}\frac{1}{KM}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{j=0}^{1}\Bigg[\pi\left([a_t^i]_{mkj}|s_t^i\right) \times \Big[Q_\theta\left([a_t^i]_{mkj}, s_t^i\right) - \psi\log\left(\pi([a_t^i]_{mkj}|s_t^i)\right)\Big]\Bigg],$$

where the value of Q_θs appearing in the gradient is substituted by the minimum of two CNs.

14. Update TCNs by the weights of CNs after T_{D-SAC} steps.
15. $t \leftarrow t + 1$
16. **end**

IV. Numerical Results

Network topology consists of M APs and K users that are randomly distributed within a disk of 0.5 km radius, where it is assumed that $M \in [10, 60]$, and $K = 10$ in most simulations unless otherwise specified in the text. We refer to every random topology generated as above by a positioning instance. Random pilot assignment is utilized where every user is randomly assigned a pilot sequence from a pool of τ_p orthogonal pilots of length τ_p. Furthermore, λ in (22) is selected heuristically as $\lambda = 10^6$. Large-scale fading coefficients are modeled as in [27]

$$\beta_{mk} = 10^{\frac{PL(d_{mk}) + \sigma_{sh} z_{mk}}{10}}, \tag{29}$$

where $z_{mk} \sim N(0, 1)$ is the shadow fading random variable with standard deviation $\sigma_{sh} = 8$ dB. Furthermore, $PL(d_{mk})$ is the three-slope path-loss model in dB which is given by [27], [34]

$$PL(d_{mk}) = -140.7 - 35\log_{10}(d_{mk}) + 20c_0\log_{10}\left(\frac{d_{mk}}{d_0}\right)$$

TABLE I: Simulation parameters.

System bandwith, B	20 MHz
Coherence interval, τ_c	200 symbols
Length of upling training, τ_p	5 symbols
Normalized SNR of each pilot symbol, ρ_p	$0.2/N_0$
Pilot power control coefficient, $\zeta_k, \forall k$	1
Normalized uplink SNR, ρ_u	$1/N_0$
Data power control coefficient, $\eta_k, \forall k$	1
Refrence distances, (d_0, d_1)	(10,50)m
Noise power at receivers, N_0	-104 dBm
Rate threshold, R_{th}	0.5 bits/s/Hz
Power amplifier efficiency, $\nu_k, \forall k$	0.4
Internal power consumption, $P_{tc,m}, \forall m$	0.2 W
Fixed power consumption, $P_{fix,m}, \forall m$	0.825 W
Traffic-dependent power, $P_{ft,m}$	0.25 W/(Gbits/s)

$$+ 15c_1 \log_{10}\left(\frac{d_{mk}}{d_1}\right),$$

where d_0 and d_1 denote two reference distance in km, and d_{mk} represents the distance between m'th AP and k'th user in km. Other system parameters are selected as in Table I, which are similar to [7], [3].

Table II shows the selected hyperparameter values for each of the three reinforcement learning algorithms. Given DQL poor performance, it has been omitted from all figures. Also, we set the value of discount factor, η to 0.99, and batch size $\tilde{N}$ to 512 for all proposed algorithms. Given the exorbitant processing demand i.e., power and time of optimal hyperparameter selection schemes [35], our hyperparameters were selected heuristically. In addition, the rectified linear unit (ReLu) activation function is used in each layer of all DNNs, and the Adam optimizer is applied to update the DNNs weights in all algorithms. As shown in Table II, 5 hidden layers, with 512 neurons in each layer, are considered for DDPG and D-SAC. Different number of layers is selected for various algorithms, because DDPG and D-SAC with 5 layers outperformed a 7 layer setup, while C-SAC with 7 layers outperformed a 5 layer setup.

Apart from the three DRL algorithms, namely DDPG, C-SAC, and D-SAC, the following three classic methods are implemented for comparison:

i. **All-AP:** In this method, all the M APs serve all the K users without considering APS at all.

ii. **Delta-APS:** The k-th user is served by $M_k \leq M$ APs maintaining the M_k largest large-scale fading coefficients [3]. M_k is selected as the smallest natural number that satisfies

$$\frac{\sum_{m=1}^{M_k} \overline{\beta}_{mk}}{\sum_{m=1}^{M} \beta_{mk}} \geq \delta, \tag{30}$$

where $\{\overline{\beta}_{1k}, ..., \overline{\beta}_{Mk}\}$ denote the sorted set of $\{\beta_{1k}, ..., \beta_{Mk}\}$ in descending order, and δ is a known constant denoting the presumed percentage of the total channel power which is exploited for each user. In order to select the best value of δ, we consider a set of δ_is from zero to one with 0.1 increments. For each δ_i, Delta-APS is applied and the energy efficiency is calculated. Finally, the δ_i which can meet the R_{th} for a larger number of users and has the highest energy efficiency is selected. In the so-called "Ideal" Delta-APS, the best value for δ is separately evaluated for each different positioning instance. However, in the "Average" Delta-APS, a fixed average value of best δs is utilized for all positioning instances. While Ideal Delta APS performs better, Average Delta-APS maintains a lower complexity.

iii. **N-APS:** Each user selects $N \leq M$ APs whose channels have the largest gains. To select the best value of N, we consider a set of N_is from 1 to M. For each N_i, N-APS is performed and the corresponding energy efficiency is calculated. Finally, the N_i is selected which can meet the R_{th} for a larger number of users and has the highest energy efficiency. In the "Ideal" N-APS, the best value of N is selected differently for each positioning instance. However, in the "Average" N-APS, the value of N is considered fixed for all positioning instances and this fixed value is selected as the average of the best values of N for each positioning instance. While Ideal N-APS performs better, Average N-APS maintains a lower complexity.

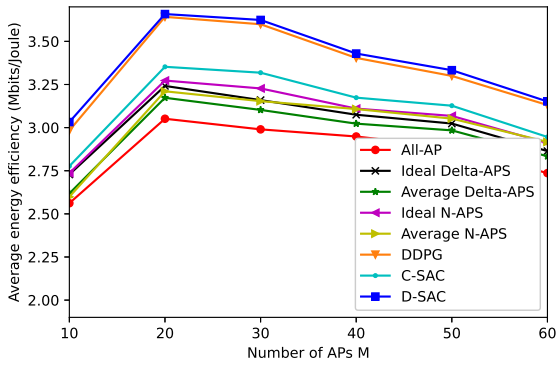

Fig. 2: Average energy efficiency w.r.t. the number of APs

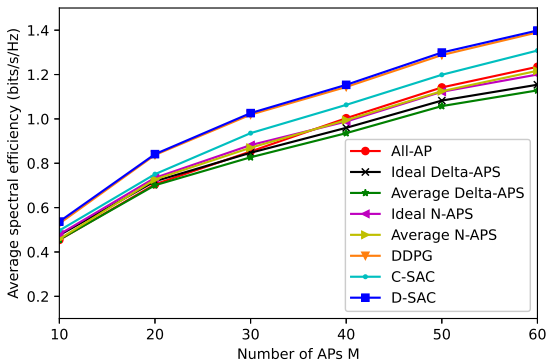

Fig. 3: Average spectral efficiency w.r.t. the number of APs

A. *Low-Velocity Users*

In this scenario, the location of users are considered constant over time. Average performance over 100 positioning instances is plotted. Because for each positioning instance, user locations are fixed, states s_t and next state s_{t+1} are equal in all DRL algorithms. Fig. 2 and Fig. 3 illustrate the average EE and SE versus the number of APs, respectively. Evidently, as the number of APs increases, average SE increases since users can

TABLE II: Hyper parameters of DRL algorithms.

DDPG	Number of AN, CN, TAN and TCN fully connected hidden layers	5
	Number of neurons per layers	(512,512,512,512,512)
	AN learning rate $\lambda_{\mathrm{AN-DDPG}}$	0.0001
	CN learning rate $\lambda_{\mathrm{CN-DDPG}}$	0.0001
	V noise variance Υ	0.0001
	Soft update parameter κ	0.9
C-SAC	Number of AN, CN, and TCN fully connected hidden layers	7
	Number of neurons per layer	(256,256,512,512,512,256,256)
	Entropy target	$-\lvert A\rvert$
	Learning rate $\lambda_{\mathrm{C-SAC}}$	0.0001
	Target update $T_{\mathrm{C-SAC}}$	100
D-SAC	Number of AN, CN, and TCN fully connected hidden layers	5
	Number of neurons per layer	(512,512,512,512,512)
	Entropy target	$-0.98\log(\frac{1}{\lvert A\rvert})$
	Learning rate $\lambda_{\mathrm{D-SAC}}$	0.0001
	Target update $T_{\mathrm{D-SAC}}$	100

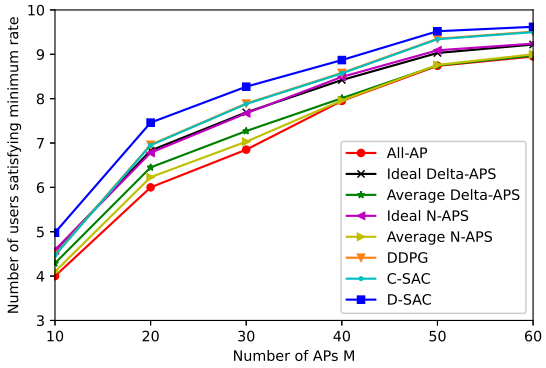

Fig. 4: Average number of users whose rate is above the threshold

be served by more APs. At the same time, EE first increases, and then decreases because of the power consumption growth. As depicted in Fig. 2, the Delta-APS and N-APS in both ideal and average versions improve EE compared to All-AP due to their selective AP strategy. However, our proposed algorithms DDPG, C-SAC, and D-SAC outperform these classic methods. In addition, as revealed in Fig. 3, Delta-APS and N-APS do not improve SE of the system since they do not consider the interference between various users in their selection strategy. On the other hand, DDPG, C-SAC, and D-SAC improve both EE and SE simultaneously in spite of the fact that they are designed with only EE in mind. Among the proposed DRL algorithms, D-SAC and DDPG have almost the same performance, and outperform C-SAC. In Fig. 2 and Fig. 3, the gap between C-SAC and D-SAC is approximately 0.3×10^6 and 0.1, respectively.

Finally, the number of users whose minimum rate constraint R_{th} are satisfied for the value of $\lambda = 10^6$ is plotted in Fig. 4 versus the number of APs. Given that none of the existing approaches in Fig. 4 can guarantee minimum rate for all users, we also fixed the value of λ heuristically. We should note that as $\lambda \to \infty$, our algorithms can guarantee QoS for all users. However, to avoid numerical instability we fixed λ to 10^6 and plotted the average number of devices whose QoS are satisfied. As shown in Fig. 4, by increasing the number of APs, the number of users meeting R_{th} also increases. The Average Delta APS and Average N-APS are capable of serving more users than All-AP for $M \in [10, 40]$, and the Ideal versions perform even better. Again, the proposed DRL algorithms DDPG, C-SAC, and D-SAC outperform the classic methods. Among the DRL algorithms, DDPG and C-SAC serve almost the same number of users, while D-SAC outperforms all other methods.

In all simulated DRL algorithms, in order to improve exploration, the improvement approach in [33] is utilized. Reference [33] suggests that before starting the learning process for each of the DRL algorithms, T random actions are generated and their corresponding EE is evaluated and stored in replay memory as experiences. For this purpose, T positioning instances as states and corresponding T random actions are generated. Then, based on the given states and actions, rewards i.e., EEs are calculated. Parameter T is set to 20000 in all simulations. It is noteworthy that this random data generation and collection can be performed offline. Hence, it does not adversely affect algorithm delay in its online stage.

To address practical concerns regarding how DRL convergence delays can adversely affect performance, three remarks will be made.

Remark 1: Since the input to all DRL algorithms are large-scale fading coefficients, the DRL convergence delay can be on the order of a fraction of large-scale coherence time. For low-velocity users, this large-scale coherence time can be very large. As a result, by assuming a high-speed CPU or the possibility of performing the DRL algorithms in the cloud, convergence time can be dragged below a satisfactory limit. It is notable that each DRL iteration amounts to EE evaluation and does not need any interaction between APs and users. As a result, DRL algorithms can converge in reasonable time.

Remark 2: As a second alternative, before the convergence of the DRL algorithm, one can apply the sub-optimal Ideal N-APS, which has the best performance among classic methods in the beginning of each large-scale coherence time. As soon as the DRL algorithm converges, its APS strategy can replace that of Ideal N-APS.

Remark 3: Upon assuming that the channel model is not block-fading but is slowly varying over time, the tracking ability of DRL algorithms can be utilized to our advantage. This means that after convergence, DRL algorithm can track the changes in optimal APS strategy in real-time as large-scale fading coefficients change slowly.

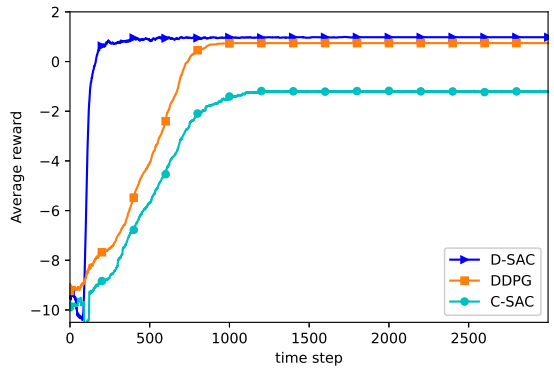

Fig. 5: Average collected reward w.r.t. time

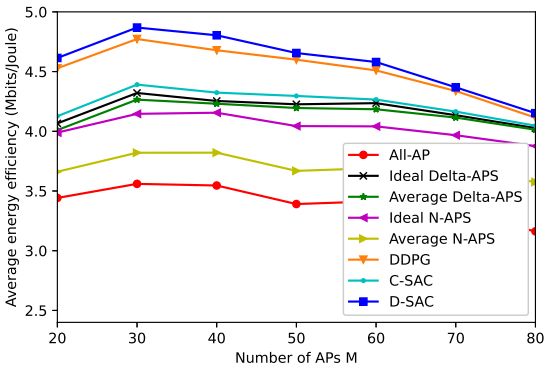

Fig. 6: Average Energy efficiency w.r.t. number of APs

B. Multiple Antenna APs and 3GPP Urban Microcell Model

Here, we consider $O = 2$ antennas per AP, a 3GPP Urban Microcell model [36], and a denser network compared to Subsection IV.A. Our aim is to demonstrate the competitiveness of our proposed DRL algorithms over various scenarios. Network topology consists of $M \in [20, 80]$ APs and $K = 20$ users that are randomly distributed within a disk of 0.5 km radius. Every user is randomly assigned an orthogonal pilot sequence from a pool of $\tau_p = 10$ pilots of length $\tau_p = 10$. Large-scale fading coefficients are modeled as in [36]

$$\beta_{mk} = -30.5 - 36.7 \log_{10}\left(\frac{d_{mk}}{1^m}\right) + F_{mk}, \qquad (31)$$

where β_{mk}s are in dB, d_{mk} is the distance between k'th user and m'th AP in meters, and $F_{mk} \sim N(0, 4^2)$ represents the shadow fading. Shadowing between an AP m to user k and AP i to user j are correlated as

$$\mathbb{E}\{F_{mk}F_{ij}\} = \begin{cases} 4^2 \times 2^{-\delta_{kj}/9^m} & m = i, \\ 0 & m \neq i, \end{cases} \qquad (32)$$

where δ_{kj} is the distance between k'th user and j'th user. The first row in (32) denotes the correlation of shadowing between a single AP and two different users. Second row in (32) accounts for the correlation of shadowing between two different APs and arbitrary users, which is negligible since we assumed a separation of at least 50 meters between adjacent APs leading to $2^{-50/9} = 0.02 \approx 0$. Given that we are still investigating low-velocity users, the location of users are considered constant over time. Average performance over 100 positioning instances is plotted.

To the best of our knowledge, a rigorous proof for the convergence of model-free actor-critic methods or any RL approach that utilizes neural network approximations is an open problem, see e.g., [30, p. 241]. Therefore, we have illustrated the convergence of our proposed DRLs numerically for the given parameters, and plotted the collected reward versus time in Fig. 5. It can be observed that all three modified algorithms converge in terms of average collected reward. We have set $M = 30$ and $K = 20$ in Fig. 5. According to Fig. 5, D-SAC, DDPG, and C-SAC converge after approximately 250, 1000, and 1250 time steps, respectively.

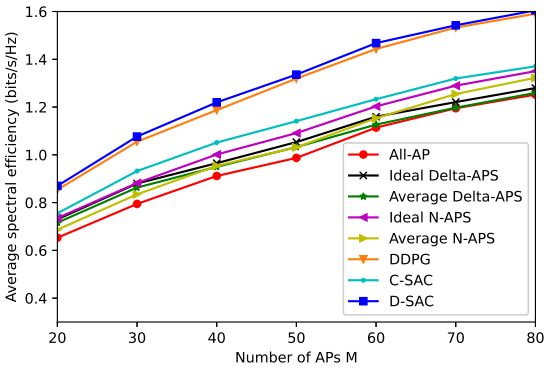

Fig. 7: Average spectral efficiency w.r.t. number of APs

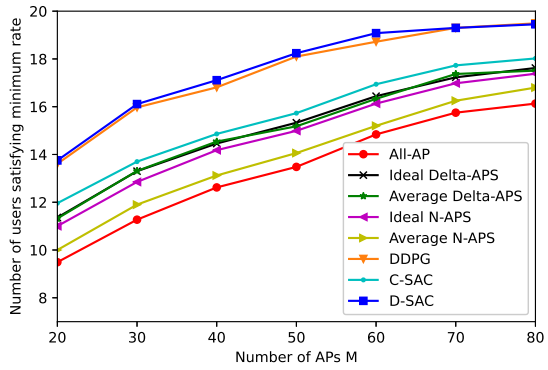

Fig. 8: Average number of users whose rate is above the threshold

Average performance in terms of EE and SE are plotted in Figs. 6 and 7 respectively. According to these figures, D-SAC and DDPG clearly outperform existing approaches, while C-SAC maintains a small but visible gain over the alternatives. The number of users whose minimum rate constraint are satisfied for the value of $\lambda = 10^6$ is plotted in Fig. 8. Again, it can be observed that D-SAC/DDPG, and C-SAC outperform existing approaches by significant and visible margins respectively.

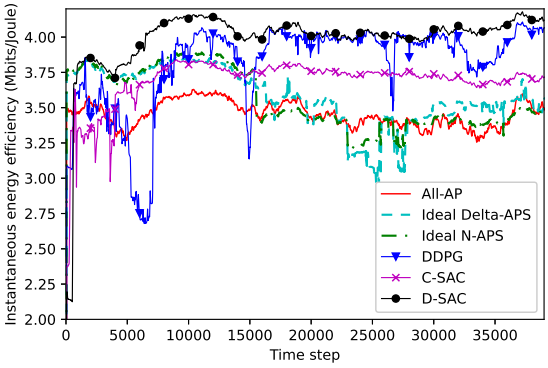

Fig. 9: Energy efficiency w.r.t. time

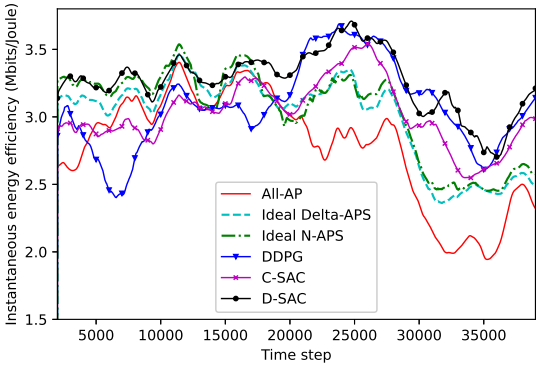

Fig. 11: Energy efficiency w.r.t. time

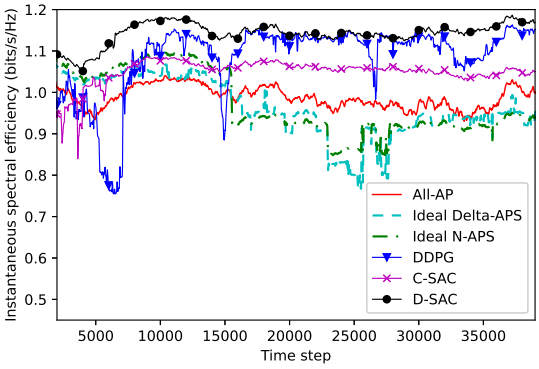

Fig. 10: Spectral efficiency w.r.t. time

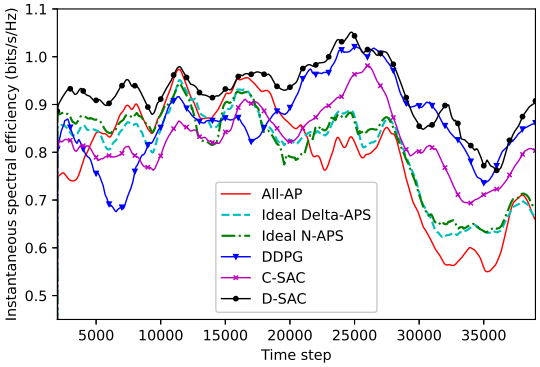

Fig. 12: Spectral efficiency w.r.t. time

C. High-Velocity Users

To model user movements, two different scenarios are considered. In the first scenario, an initial location and a velocity in range of [0,100] km/h are selected for each user. It is assumed that each user moves from its initial location with the predetermined velocity in a direction, which is randomly selected from the range of $[0,2\pi]$. The selected velocity of users remains constant along the trajectory. In each coherence time τ_c, the direction of users is changed randomly. It is also considered that if the user reaches the edge of the environment, the direction of its movement will change π degrees. In this scenario the state s_t and the next state s_{t+1} are not equal, since user moves along trajectory over time. The number of APs M is considered equal to 30.

In this scenario, the proposed DRL algorithms DDPG, C-SAC and D-SAC are compared with the aforementioned All-AP, Ideal Delta-APS, and Ideal N-APs in 40000 time steps. Fig. 9 and Fig. 10 illustrate a 50-wide moving average of the instantaneous EE and SE versus time steps. Observations indicate that the proposed DRL algorithms begin with weak performance and continuously improve over time. After approximately 1000 time steps, D-SAC outperforms other methods. DDPG also outperforms Ideal N-APS after 8000 time steps and reaches the same level as the performance of D-SAC after 17000 time steps. However, it is observed that DDPG has greater fluctuations than D-SAC and suffers a sharp performance drop in the 15000 and 27000 time steps. Finally, after 15000 time steps, C-SAC outperforms Ideal N-APS but is less energy efficient than D-SAC.

In the second scenario, most settings are taken to be similar to the first setup. However, the direction of each user's movement does not change in every time step. Instead, direction changes randomly every 200 time steps. As a result, the location of users changes more rapidly than the first scenario. The number of APs M equals to 30.

Fig. 11 and Fig. 12 illustrate a 2000-wide moving average of the instantaneous EE and SE versus time step. The same behavior as the first scenario is observed. D-SAC outperforms other methods from the beginning, and after about 20000 time steps a visible gap exists between D-SAC and Ideal N-APS. C-SAC and DDPG methods begin with a weak performance, and they constantly improve over time. After about 18000 time steps, DDPG outperforms Ideal N-APS and comes close to the D-SAC. C-SAC outperforms N-APS after 20000 time steps and is less efficient than D-SAC and DDPG.

V. Conclusion

We have investigated the problem of AP-user association for uplink CF massive MIMO, where we incorporate all sources of practical limitation such as erroneous CSI at the APs, pilot contamination during training, and statistical CSI at the CPU. We have proposed modified DRL algorithms that can tackle this problem with both satisfactory performance and complexity. While D-SAC, DDPG, and C-SAC performed satisfactorily, DQL was revealed to perform poorly and even

worse than heuristic methods. These observations suggest that while DRL approaches usually perform superior to existing prior art, a wrong choice of DRL algorithm, e.g., DQL, can have a large negative impact on their performance. As future directions, we can utilize a continuous approximation of discrete constraints to reach a possibly high quality sub-optimal solution via derivative-based optimization methods.

REFERENCES

[1] E.-K. Hong, I. Lee, B. Shim, Y.-C. Ko, S.-H. Kim, S. Pack, K. Lee, S. Kim, J.-H. Kim, Y. Shin, Y. Kim, and H. Jung, "6G R&D vision: Requirements and candidate technologies," *Journal of Communications and Networks*, vol. 24, pp. 232–245, April 2022.

[2] M. Alonzo, S. Buzzi, A. Zappone, and C. D'Elia, "Energy-efficient power control in cell-free and user-centric massive MIMO at millimeter wave," *IEEE Transactions on Green Communications and Networking*, vol. 3, pp. 651–663, Sept. 2019.

[3] H. Q. Ngo, L.-N. Tran, T. Q. Duong, M. Matthaiou, and E. G. Larsson, "On the total energy efficiency of cell-free massive MIMO," *IEEE Transactions on Green Communications and Networking*, vol. 2, pp. 25–39, March 2018.

[4] H. T. Dao and S. Kim, "Effective channel gain-based access point selection in cell-free massive MIMO systems," *IEEE Access*, vol. 8, pp. 108127–108132, June 2020.

[5] S. Buzzi, C. D'Andrea, A. Zappone, and C. D'Elia, "User-centric 5G cellular networks: Resource allocation and comparison with the cell-free massive MIMO approach," *IEEE Transactions on Wireless Communications*, vol. 19, pp. 1250–1264, Feb. 2020.

[6] T. Van Chien, E. Björnson, and E. G. Larsson, "Joint power allocation and load balancing optimization for energy-efficient cell-free massive MIMO networks," *IEEE Transactions on Wireless Communications*, vol. 19, pp. 6798–6812, Oct. 2020.

[7] H. V. Nguyen, V.-D. Nguyen, O. A. Dobre, S. K. Sharma, S. Chatzinotas, B. Ottersten, and O.-S. Shin, "On the spectral and energy efficiencies of full-duplex cell-free massive MIMO," *IEEE Journal on Selected Areas in Communications*, vol. 38, pp. 1698–1718, Aug. 2020.

[8] G. Femenias, N. Lassoued, and F. Riera-Palou, "Access point switch on/off strategies for green cell-free massive MIMO networking," *IEEE Access*, vol. 8, pp. 21788–21803, Jan. 2020.

[9] J. García-Morales, G. Femenias, and F. Riera-Palou, "Energy-efficient access-point sleep-mode techniques for cell-free mmWave massive MIMO networks with non-uniform spatial traffic density," *IEEE Access*, vol. 8, pp. 137587–137605, July 2020.

[10] T. X. Vu, S. Chatzinotas, S. ShahbazPanahi, and B. Ottersten, "Joint power allocation and access point selection for cell-free massive MIMO," in *Proc. IEEE International Conference on Communications (ICC)*, pp. 1–6, July 2020.

[11] H. Q. Ngo, H. Tataria, M. Matthaiou, S. Jin, and E. G. Larsson, "On the performance of cell-free massive MIMO in Ricean fading," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, pp. 980–984, Feb. 2018.

[12] C. D'Andrea and E. G. Larsson, "User association in scalable cell-free massive MIMO systems," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, pp. 826–830, Nov. 2020.

[13] M. Guenach, A. A. Gorji, and A. Bourdoux, "Joint power control and access point scheduling in fronthaul-constrained uplink cell-free massive MIMO systems," *IEEE Transactions on Communications*, vol. 69, no. 4, pp. 2709–2722, 2021.

[14] S. Hwang, H. Kim, H. Lee, and I. Lee, "Multi-agent deep reinforcement learning for distributed resource management in wirelessly powered communication networks," *IEEE Transactions on Vehicular Technology*, vol. 69, pp. 14055–14060, Nov 2020.

[15] S. Biswas and P. Vijayakumar, "AP selection in cell-free massive MIMO system using machine learning algorithm," in *Proc. International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 158–161, May 2021.

[16] M. Guenach, A. Gorji, and A. Bourdoux, "A deep neural architecture for real-time access point scheduling in uplink cell-free massive MIMO," *IEEE Transactions on Wireless Communications*, pp. 1–1, Dec. 2021.

[17] V. Ranasinghe, N. Rajatheva, and M. Latva-aho, "Graph neural network based access point selection for cell-free massive MIMO systems," in *Proc. Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec. 2021.

[18] X. Chai, H. Gao, J. Sun, X. Su, T. Lv, and J. Zeng, "Reinforcement learning based antenna selection in user-centric massive MIMO," in *Proc. Vehicular Technology Conference (VTC)*, pp. 1–6, June 2020.

[19] C. F. Mendoza, S. Schwarz, and M. Rupp, "Deep reinforcement learning for dynamic access point activation in cell-free MIMO networks," in *Proc. ITG Workshop on Smart Antennas*, pp. 1–6, Nov. 2021.

[20] F. Fredj, Y. Al-Eryani, S. Maghsudi, M. Akrout, and E. Hossain, "Distributed beamforming techniques for cell-free wireless networks using deep reinforcement learning," *IEEE Transactions on Cognitive Communications and Networking*, April 2022 (Early Access).

[21] R. Y. Chang, S.-F. Han, , and F.-T. Chien, "Reinforcement learning-based joint cooperation clustering and content caching in cell-free massive MIMO networks," in *Proc. Vehicular Technology Conference (VTC)*, Sept. 2021.

[22] Y. Al-Eryani and E. Hossain, "Self-organizing mmWave MIMO cell-free networks with hybrid beamforming: A hierarchical DRL-based design," *IEEE Transactions on Communications*, vol. 70, pp. 3169–3185, May 2022.

[23] Y. Zhong, T. Q. S. Quek, and X. Ge, "Heterogeneous cellular networks with spatio-temporal traffic: Delay analysis and scheduling," *IEEE Journal on Selected Areas in Communications*, vol. 35, pp. 1373–1386, June 2017.

[24] X. Ge, B. Yang, J. Ye, G. Mao, C.-X. Wang, and T. Han, "Spatial spectrum and energy efficiency of random cellular networks," *IEEE Transactions on Communications*, vol. 63, pp. 1019–1030, March 2015.

[25] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. USA: Prentice-Hall, Inc., 1993.

[26] T. C. Mai, H. Q. Ngo, M. Egan, and T. Q. Doung, "Pilot power control for cell-free massive MIMO," *IEEE Transactions on Vehicular Technology*, vol. 67, pp. 11264–11268, Nov 2018.

[27] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Transactions on Wireless Communications*, vol. 16, pp. 1834–1850, March 2017.

[28] B. Hassibi and B. Hochwald, "How much training is needed in multiple-antenna wireless links?," *IEEE Transactions on Information Theory*, vol. 49, pp. 951–963, July 2003.

[29] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, second ed., 2018.

[30] D. P. Bertsekas, *Reinforcement learning and optimal control*. Athena Scientific, 2019.

[31] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," *arXiv preprint*, Dec. 2019.

[32] 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 preprint arXiv:1509.02971*, July 2015.

[33] P. Christodoulou, "Soft actor-critic for discrete action settings," *arXiv preprint*, 2019.

[34] A. Tang, J. Sun, and K. Gong, "Mobile propagation loss with a low base station antenna for NLOS street microcells in urban area," in *Proc. Vehicular Technology Conference (VTC)*, vol. 1, pp. 333–336, May 2001.

[35] R. Liessner., J. Schmitt., A. Dietermann., and B. Bäker., "Hyperparameter optimization for deep reinforcement learning in vehicle energy management," in *Proc. of International Conference on Agents and Artificial Intelligence*, pp. 134–144, SciTePress, Feb. 2019.

[36] E. Björnson and L. Sanguinetti, "Making cell-free massive MIMO competitive with MMSE processing and centralized implementation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 77–90, 2020.

Niyousha Ghiasi was born in Tehran, Iran in 1996. She received her B.Sc. and M.Sc. degrees in electrical engineering from Alzahra University and Iran University of Science and Technology (IUST), Tehran, Iran, in 2018 and 2022, respectively. She has also been with the Mobile Broadband Network Research Group (MBNRG) at IUST since 2018. Her research interests mainly lie in the area of machine learning in wireless communications and networking.

This article has been accepted for publication in IEEE Transactions on Green Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TGCN.2022.3196013

Shima Mashhadi was born in Arak, Iran in 1996. She received her B.Sc. and M.Sc. degrees in electrical engineering from Alzahra University and Iran University of Science and Technology (IUST), Tehran, Iran, in 2018 and 2022, respectively. She has also been with the Mobile Broadband Network Research Group (MBNRG) at IUST since 2018. Her research interests mainly lie in the area of machine learning in wireless communications and networking.

Shahrokh Farahmand was born in Tehran, Iran in 1980. He received his B.Sc. degree in electrical engineering from Sharif University of Technology in 2003. Then, he pursued his graduate studies in United States were he obtained his M.Sc. degree in 2006 and Ph.D. degree in 2011 both from University of Minnesota (UMN) at Twin-Cities in the field of communications and signal processing. From 2011 to 2014 he was with Iran Research Organization for Science and Technology (IROST) where he held a research faculty position. Since 2018, he has been with the electrical engineering department at Iran University of Science and Technology (IUST) where he is currently an assistant professor. His general interests include applications of statistical signal processing, optimization, and machine learning in communications and networking. His current focus is on internet of things (IoT), intelligent reflecting surfaces (IRS), massive MIMO, and ultra-wideband impulse radio (UWB-IR).

S. Mohammad Razavizadeh received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the Iran University of Science and Technology (IUST), Tehran, Iran, in 1997, 2000, and 2006, respectively. From June 2004 to April 2005, he was a Visiting Researcher with the Coding and Signal Transmission Laboratory, University of Waterloo, ON, Canada. From 2005 to 2011, he was with the Iran Telecommunication Research Center, as a Research Assistant Professor. Since 2011, he has been with the School of Electrical Engineering, IUST, where he is currently an Associate Professor. He was also a Visiting Professor with Korea University, South Korea and Chalmers University, Sweden. His research interests are in the area of signal processing for wireless communication systems and cellular networks. He is a Senior Member of IEEE.

Inkyu Lee received the B.S. degree (Hons.) in Control and Instrumentation Engineering from Seoul National University, Seoul, South Korea, in 1990, and the M.S. and Ph.D. degrees in Electrical Engineering from Stanford University, Stanford, CA, USA, in 1992 and 1995, respectively. From 1995 to 2002, he was a Member of the Technical Staff with Bell Laboratories, Lucent Technologies, Murray Hill, NJ, USA, where he studied high-speed wireless system designs. Since 2002, he has been with Korea University, Seoul, where he is currently a Professor with the School of Electrical Engineering. He has also served as the Department Head of the School of Electrical Engineering, Korea University, from 2019 to 2021. In 2009, he was a Visiting Professor with the University of Southern California, Los Angeles, CA, USA. He has authored or coauthored more than 200 journal articles in IEEE publications and holds 30 U.S. patents granted or pending. His research interests include digital communications, signal processing, and coding techniques applied for next-generation wireless systems. He was elected as a member of the National Academy of Engineering of Korea in 2015. He was a recipient of the IT Young Engineer Award at the IEEE/IEEK Joint Award in 2006, the Best Paper Award at the Asia–Pacific Conference on Communications in 2006, the IEEE Vehicular Technology Conference in 2009, the Best Research Award from the Korean Institute of Communications and Information Sciences in 2011, the IEEE International Symposium on Intelligent Signal Processing and Communication Systems in 2013, the Best Young Engineer Award from the National Academy of Engineering of Korea in 2013, and the Korea Engineering Award from the National Research Foundation of Korea in 2017. He served as an Associate Editor for the IEEE Transactions on Communications from 2001 to 2011 and the IEEE Transactions on Wireless Communications from 2007 to 2011. In addition, he was a Chief Guest Editor of the IEEE Journal on Selected Areas in Communications Special Issue on “4G wireless system” in 2006. He also serves as the Co-Editor-in-Chief for the Journal of Communications and Networks. He is also an IEEE Fellow and a Distinguished Lecturer of IEEE.