Joint beamforming and user association with reduced CSI signaling in mobile environments: A Deep Q-learning approach

Ha Duc Thang, Lila Boukhatem, Megumi Kaneko, Nhan Nguyen-Thanh

Version of Record: https://www.sciencedirect.com/science/article/pii/S1389128621003078
Manuscript_fbf3239f44c055d99e2d8b5f69dc03f0

Joint Beamforming and User association with Reduced CSI Signaling in Mobile Environments: a Deep Q-Learning Approach

Ha Duc Thang*, Lila Boukhatem*, Megumi Kaneko[2], Nhan Nguyen-Thanh *
LRI Laboratory, CNRS/Univ. Paris Saclay, Orsay, France*
National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, 101-8430 Tokyo, Japan [2]

***Abstract*—Heterogeneous Cloud Radio Access Network (H-CRAN) is considered as a cost-efficient network solution to meet 5G data traffic requirements. In this paper, we consider the problem of beamforming and user clustering (user-to-Remote Radio Head (RRH) association) in the downlink of a H-CRAN where users have heterogeneous mobility profiles. Given the rapidly time-varying nature of the mobile wireless environment, it is challenging to offer an optimal beamforming and user association performance during a long-term allocation process without incurring large Channel State Information (CSI) and signaling overheads. For that purpose, we proposed in [1] an Adaptive Beamforming and User Clustering (ABUC) algorithm which resolves the joint beamforming and user clustering problem when considering CSI cost and imperfectness under user mobility assumptions. In this paper, we design a deep reinforcement-learning framework which enables the proposed ABUC algorithm to select on-the-fly its best scheduling parameters, namely the period and type of CSI feedback, given each user mobility profile. The proposed ABUC-DQL approach can overcome the scalability limitation of the Q-learning approach [1] and better handle the problem when formulated using a POMDP (Partially Observable Markov Decision Process) model. The simulation results show that the convergence time is mainly impacted by the number of users in the network, and the online-learning ability of the framework can quickly adapt to the changes of users mobility.**

***Index Terms*—H-CRAN, beamforming, user-to-RRH association, deep Q-learning**

I. Introduction

The fifth generation (5G) of wireless communication systems is expected to embrace the unremitting exponential growth of mobile data traffic and the stringent QoS (Quality of Service) demands of a large panel of new emerging services and applications. 5G is also envisioned to overcome several challenges such as the heterogeneous deployment of cells (macro and small cells), and the ultra-dense users environments with varying mobility profiles and behaviours [2].

To achieve higher efficiency, 5G systems rely on a combination of advanced technologies such as millimetre wave, cloud computing, network slicing and small cells, etc. By taking full advantage of the cloud computing and heterogeneous and small cells deployments [3], [4], the Heterogeneous Cloud Radio Access Network (H-CRAN) is considered as one of the most promising architectures to support 5G and beyond [3] (Fig. 1). In H-CRAN, the macrocells and small cells (picocells, microcells, femtocells) are densely deployed low power Remote Radio Heads (RRHs) forming Macro-RRHs or Pico-RRHs. The baseband signals of users are relayed through the capacity-limited fronthaul links to be processed by the centralized Baseband Unit (BBU) Pool.

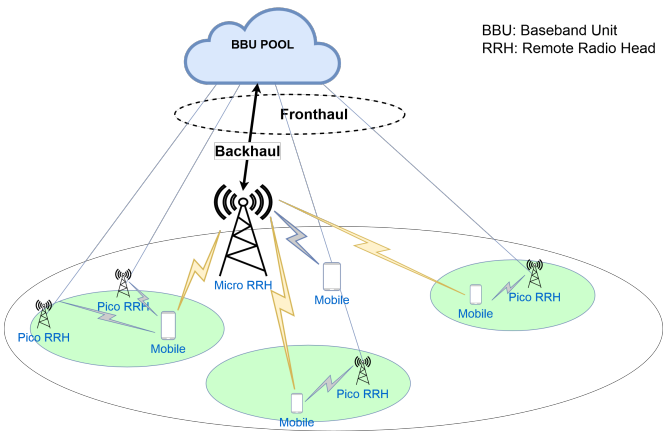


Fig. 1: H-CRAN system model

To guarantee optimal spectrum reuse across all RRHs, the system requires advanced radio resource management mechanisms such as intercell interference coordination, Coordinated Multi-Point (CoMP) in Mutiple Input Multiple Output (MIMO), Beamforming (BF) and optimal user clustering (user-to-RRH association). Several research works have addressed the interference mitigation problem in MIMO channels when considering the weighted sumrate optimization [5], [6], [7]. Other works tackled the same problem under fronthaul constraint considerations [8], [9], [10]. Some studies have focused on either the sum-rate, Spectral Efficiency (SE), or Energy Efficiency (EE) maximization problems by proposing the joint optimization of beamforming and user clustering [11], [12], [13], [14].

More recent efforts have used reinforcement learning approaches and more particularly Deep Reinforcement Learning (DRL). DRL has been applied in advanced radio environments of future 6G systems such as the reconfigurable intelligent surface (RIS) to further improve the massive MIMO transmissions. The authors in [15] investigated the joint design of transmit beamforming matrix at the base station and the phase shift matrix at the RIS to maximize the sum rate of multiuser

downlink MISO (multiple input single output) systems by utilizing DRL. In [16], a DRL-based hybrid beamforming scheme for multi-hop RIS-assisted communications is proposed to improve the coverage range at TeraHertz band frequencies. The DRL method proved its effectiveness in solving the non-convex joint design problem of the digital beamforming at the base station and analog beamforming matrices at the RISs.

Other research works, have focused on Resource Allocation (RA) problems. The authors in [17] addressed RA and interference management problems and proposed cache enabled opportunistic interference alignment (IA) in which the caching avoided the CSI exchange and hence alleviated the burden of signalling on the backhaul. In [18], DRL was used to propose a reduced-complexity joint optimization of radio resources, caching and computing resources. The DRL-based approach has also been used for different purposes in several works: to propose a resource management algorithm for Fog Radio Access Networks (FRAN) in [19], develop a dynamic resource allocation for CRANs in [20], and in [21], to design a distributed dynamic power allocation scheme for weighted sum-rate maximization.

However, these solutions suffer from high computational complexity and generally rely on perfect Channel State Information (CSI) knowledge, thereby requiring a large amount of control signaling and CSI overhead. Furthermore, most of these works did not consider the influence of user mobility and the resulting time-varying wireless environment over the long-term scheduling performance. To overcome these limitations which are magnified under high mobility scenarios, we have proposed in [1] a mobility-aware Adaptive Beamforming and User Clustering (ABUC) algorithm based on a Q-Learning (QL) approach. The proposed learning algorithm allows to wisely tune the various solution parameters acting on the global network performance, for instance CSI feedback parameters (periodicity and type). However, this preliminary solution is only applicable in a small network scenario with a limited number of users. This is because Q-learning is not appropriate for solving problems with large spaces of actions and states.

In this work, we propose an optimized radio resource allocation framework in downlink H-CRAN which, unlike other traditional reference solutions, is able to alleviate the control and signaling costs while coping with heterogeneous users mobility. To do so, we address the joint beamforming and user clustering optimization problem. More precisely, we formulate this as a weighted sum-rate maximization problem under fronthaul capacity and maximum power budget constraints over each RRH. Moreover, our proposed framework is based on an optimization approach exploiting Deep Q-Learning (DQL) technique which is amenable to the dynamics and the complex features of real systems [22]. Extensive simulation results show that our DQL framework is able to adapt the CSI feedback parameters to the changes of users velocity. The achieved performance in terms of sum-rate, signaling costs, complexity, online learning, and convergence speed prove the efficiency of the proposed DQL framework in both homogeneous and heterogeneous mobility scenarios.

The rest of this paper is organized as follows: Section II describes the system model, the problem formulation and our reference ABUC algorithm. Section III introduces some backgrounds on Deep Q-Learning and Partially Observable Markov Decision Process (POMDP) model. We detail our proposed ABUC Deep Q-learning algorithm when assuming a POMDP problem in section IV and present the simulation results in section V. Finally, section VI concludes the paper.

II. System Model, Problem Formulation and Reference ABUC scheme

A. H-CRAN System Model

We consider the H-CRAN model in [12] which consists of a BBU Pool, L macro and pico RRHs and K users. Each RRH and user are equipped with M and N antennas, respectively, and users are randomly located in the network area.

Let $\mathbb{L} = \{1, 2, ..., L\}$ and $\mathbb{K} = \{1, 2, ..., K\}$ be the sets of RRHs and of users, respectively. The propagation channel from RRHs to the k^{th} user is denoted as $\mathbf{H}_k \in \mathbb{C}^{N \times ML}, \forall k \in \mathbb{K}$ which includes the path loss and Rayleigh fading effects.

In this paper, we focus on the downlink transmission with linear beamforming technique. Firstly, we assume that the channels are correlated between consecutive scheduling frames. We denote by $\mathbf{H}_k$ the downlink channel array from all RRHs to user k,

$$\mathbf{H}_k = \left[h_{1k}, \cdots, h_{lk}, \cdots, h_{Lk}\right]^H,$$

where h_{lk} is the channel gain from RRH l to user k.

Let $\mathbf{w}_k \in \mathbb{C}^{ML \times 1}$ be the transmit beamforming vector from all RRHs to the k^{th} user,

$$\mathbf{w}_k = \left[\mathbf{w}_{1k}^H, \cdots, \mathbf{w}_{lk}^H, \cdots, \mathbf{w}_{Lk}^H\right]^H,$$

where $\mathbf{w}_{lk} \in \mathbb{C}^{M \times 1}$.

Let $s_k \in \mathrm{C}$ be the encoded information symbol for user k with $\mathrm{E}\left[|s_k|^2\right] = 1$. The received signal at user k, $\mathbf{y}_k \in \mathbb{C}^{N \times 1}$, is expressed as

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{w}_k s_k + \mathbf{H}_k \sum_{j=1, j \neq k}^{K} \mathbf{w}_j s_j + \mathbf{n}_k,$$

where $\mathbf{n}_k \sim \mathcal{CN}(\mathbf{0}, \sigma_k^2 \mathbf{I}_N)$ is the additive white Gaussian noise and $\mathbf{I}_N$ is the identity matrix of size $N \times N$.

The Signal-to-Interference-plus-Noise Ratio (SINR) at user k can be expressed as

$$\gamma_k = \frac{\left\|\mathbf{u}_k^H \mathbf{H}_k \mathbf{w}_k\right\|^2}{\sum\limits_{j=1, j \neq k}^{K} \left\|\mathbf{u}_k^H \mathbf{H}_j \mathbf{w}_j\right\|^2 + \sigma_k^2 \|\mathbf{u}_k\|_2^2}, \tag{1}$$

where $\mathbf{u}_k \in \mathbb{C}^{N \times 1}$ is the receive beamforming vector of user k.

Assuming Minimum Mean Square Error (MMSE) decoding, the achievable rate of user k is given by

$$r_k = \log_2(1 + \mathbf{w}_k^H \mathbf{H}_k^H (\sum_{j=1, j \neq k}^{K} \mathbf{H}_k \mathbf{w}_j \mathbf{w}_j^H \mathbf{H}_k^H + \sigma_k^2 \mathbf{I}_N)^{-1} \mathbf{H}_k \mathbf{w}_k). \tag{2}$$

To model the effects of imperfect CSI knowledge at the cloud caused by user mobility, we make use of the following CSI estimation model based on [23]. Let us denote the estimated CSI matrix by $\hat{\mathbf{H}}_k \in \mathbb{C}^{N\times ML}$, $\forall k \in \mathbb{K}$ and by $\hat{h}_{nq}$, its (n,q)-*th* element, where $q=(l-1)M+m$. Hence, $\hat{h}_{nq}$ is the estimated channel gain between the m-*th* antenna of the l-*th* RRH and the n-*th* antenna of user k, expressed as

$$\hat{h}_{nq} = \lambda_k h_{nq} + (\sqrt{1-\lambda_k^2})v_{nq}. \tag{3}$$

In (3), $v_{nq} \sim \mathcal{CN}(0, F_{lk})$ where F_{lk} is the large-scale fading gain of the downlink channel from RRH l to user k, and λ_k is the correlation coefficient between $\hat{h}_{nq}$ and h_{nq} which is expressed as

$$\lambda_k = J_0(2\pi f_{d,lk} T_{dl}), \tag{4}$$

where $J_0(.)$ is the zero-*th* order Bessel function, T_{dl} is the fronthaul delay of RRH l and $f_{d,lk}$ is the maximum Doppler frequency of the channel between RRH l and user k. If user k moves at speed v_k (m/s), then the maximum Doppler frequency is calculated as $f_{d,lk} = \frac{v_k f_c}{c}$, where f_c is the carrier frequency in Hertz and c is the speed of light. Therefore, we can express λ_k as function of v_k,

$$\lambda_k = J_0\left(\frac{2\pi f_c T_{dl}}{c} v_k\right). \tag{5}$$

B. Problem Formulation

We focus on the following Weighted Sum-Rate (WSR) optimization problem [13], whereby the WSR of all users is maximized under the fronthaul link capacity constraints and individual RRH power constraints. This problem is formulated as follows,

$$\begin{aligned} \max_{\{\mathbf{w}_{lk}, l\in\mathbb{L}, k\in\mathbb{K}\}} \quad & \textstyle\sum_{k=1}^{K} \alpha_k r_k & (1)\\ \text{s.t.} \quad & P_l = \sum_{k=1}^{K} \|\mathbf{w}_{lk}\|_2^2 \le P_l^{\max} & (1a)\\ & \sum_{k=1}^{K} 1\{\|\mathbf{w}_{lk}\|_2^2\} r_k \le C_l^{\max} & (1b) \end{aligned} \tag{P1}$$

where α_k is the scheduling priority weight associated with user k. The first constraint $(1a)$ corresponds to the transmit power constraint of RRH l, i.e., P_l should be smaller than the maximum transmit power $P_l^{\max}$. The second constraint $(1b)$ expresses that the sum-rate of users connected to RRH l should be smaller than its fronthaul link capacity $C_l^{\max}$.

Problem (P1) is a non-convex mixed-integer non-linear programming (MINLP) proven to be NP-hard [13], and hence cannot be solved in polynomial time. Given its intractability, previous works [12], [13] had proposed different methods based on mathematical optimization for solving (P1). However, these methods rely on perfect CSI knowledge at the cloud, thereby incurring heavy signaling and CSI feedback costs. In the sequel, we describe our previously proposed ABUC algorithm, for sake of clarity.

C. Reference ABUC Algorithm

We have proposed in [24] the ABUC algorithm, a hybrid user clustering and beamforming scheme aiming at WSR maximization while alleviating the problem of control signaling and CSI overhead costs. ABUC is able to leverage the advantages of both dynamic and static user clusterings in H-CRAN [12], [11], where the dynamic clustering performs optimally at the expense of heavy signaling overhead, while static clustering drastically reduces the required overhead, at the expense of lower performance. However, no user mobility issues had been considered in [24], unlike our present work.

1) Dynamic Clustering Algorithm: The dynamic clustering algorithm solves the conventional WSR maximization based on a weighted minimum mean square error (WMMSE) approach that had been used in several previous works [12], [11]. The main idea is to reformulate problem (P1) into the following equivalent WMMSE problem (P2),

$$\begin{aligned} \min_{\{\mathbf{w}_{lk}, l\in\mathbb{L}, k\in\mathbb{K}\}} \quad & \textstyle\sum_k \mathbf{w}_k^H (\sum_j \alpha_j \rho_j \mathbf{H}_j^H \mathbf{u}_j \mathbf{u}_j^H \mathbf{H}_j) \mathbf{w}_k & \\ & -2\textstyle\sum_k \alpha_k \rho_k Re\{\mathbf{u}_k^H \mathbf{H}_k \mathbf{w}_k\} & (2)\\ \text{s.t.} \quad & \sum_{k=1}^{K} \|\mathbf{w}_{lk}\|_2^2 \le P_l^{\max} & (2a)\\ & \sum_{k=1}^{K} \beta_{lk} \hat{r}_k \|\mathbf{w}_{lk}\|_2^2 \le C_l^{\max}, & (2b) \end{aligned} \tag{P2}$$

where $\hat{r}_k$ denotes the rate achieved in the previous iteration. Reference [13] shows that Algorithm 1 below, referred to as the Dynamic Clustering Algorithm, enables to find a stationary point to (P2). In the reformulated constraint (2b), β_{lk} is a constant weight associated to RRH l and user k and is updated according to

$$\beta_{lk} = \frac{1}{\|\mathbf{w}_{lk}\|_2^2 + \tau}, \forall k, l, \tag{6}$$

where τ is a small constant regularization factor and $\|\mathbf{w}_{lk}\|_2^2$ is taken from the previous iteration. The corresponding Mean Square Error (MSE) is denoted as e_k,

$$e_k = \mathbf{u}_k^H (\sum_{j=1, j\neq k}^{K} \mathbf{H}_k \mathbf{w}_j \mathbf{w}_j^H \mathbf{H}_k^H + \sigma_k^2 \mathbf{I}_N) \mathbf{u}_k - 2Re\{\mathbf{u}_k^H \mathbf{H}_k \mathbf{w}_k\} + 1, \tag{7}$$

and ρ_k is the MSE weight for user k,

$$\rho_k = e_k^{-1}. \tag{8}$$

The optimal received beamforming vector $\mathbf{u}_k$ is obtained under fixed $\mathbf{w}_k$ and ρ_k,

$$\mathbf{u}_k = (\sum_{j=1, j\neq k}^{K} \mathbf{H}_k \mathbf{w}_j \mathbf{w}_j^H \mathbf{H}_k^H + \sigma_k^2 \mathbf{I}_N)^{-1} \mathbf{H}_k \mathbf{w}_k. \tag{9}$$

Algorithm 1: Dynamic Clustering Algorithm

initialize $\beta_{lk}, \hat{r}_k, \mathbf{w}_k, \forall(l,k)$
repeat
1) Fix $\mathbf{w}_k$ and compute the corresponding MSE e_k and the MMSE receiver $\mathbf{u}_k$ according to (7) and (9)
2) Update MSE weight ρ_k according to (8)
3) Compute the optimal transmit beamformer $\mathbf{w}_k$ under fixed $\mathbf{u}_k$ and ρ_k, by solving (P2)
4) Compute the achievable rate r_k
5) Update $\hat{r}_k = r_k$ and β_{kl} according to (6)
until *convergence*

2) Static Clustering Algorithm: Unlike the dynamic algorithm, in static scheduling, only a fixed subset of RRHs $\mathbb{L}_k$ serving each user k is considered. Likewise, $\mathbb{K}_l$ is defined as the subset of users associated with RRH l. Problem (P1) can be hence simplified as

$$\begin{aligned} \max_{\{\mathbf{w}_{lk}, l\in\mathbb{L}_k, k\in\mathbb{K}\}} \quad & \textstyle\sum_{k=1}^{K} \alpha_k r_k & (3) \\ \text{s.t.} \quad & P_l = \sum_{k\in\mathbb{K}_l} \|\mathbf{w}_{lk}\|_2^2 \le P_l^{\max} & (3a) \\ & \sum_{k\in\mathbb{K}_l} r_k \le C_l^{\max}, & (3b) \end{aligned} \tag{P3}$$

where constraints (3a) and (3b) involve smaller sets of users $\mathbb{K}_l$. Hence, problem (P3) can be resolved by Algorithm 2 given below, where variables $\mathbf{H}_k^{\mathbb{L}_k}$ and $\mathbf{w}_k^{\mathbb{L}_k}$ denote the channel matrix and the serving beamforming vector to user k from its RRH cluster $\mathbb{L}_k$, respectively.

Algorithm 2: Static Clustering Algorithm

initialize $\mathbb{L}_k, \beta_k, \hat{r}_k, \mathbf{w}_k, \forall k$
repeat
1) Compute (7), (8), (9) by replacing $\mathbf{w}_k$ and $\mathbf{H}_k$ by $\mathbf{w}_k^{\mathbb{L}_k}$ and $\mathbf{H}_k^{\mathbb{L}_k}$, respectively
2) Fix $\mathbb{L}_k$ during the whole process
3) Call Dynamic Clustering Algorithm to solve (P3) under fixed $\mathbb{L}_k$
until *convergence*

3) ABUC algorithm: In ABUC, we apply the dynamic and static clustering algorithms in a periodic manner as shown in Fig. 2: in the first frame of each period T referred to as a dynamic frame, the dynamic Algorithm 1 is applied, while in the remaining frames (second to T-th) referred as static frames, the static Algorithm 2 is applied under the cluster subsets fixed by the solution of the previous dynamic frame. The advantage of this approach is to consider the temporal dimension of the allocation process, while being aware of the practical feasibility of the solution in terms of complexity and signaling costs. In particular, Reference ABUC takes into account the following CSI feedback strategies during the scheduling process:

- *Full CSI knowledge:* perfect and full CSIs of all users are available at the cloud for every scheduling frame.

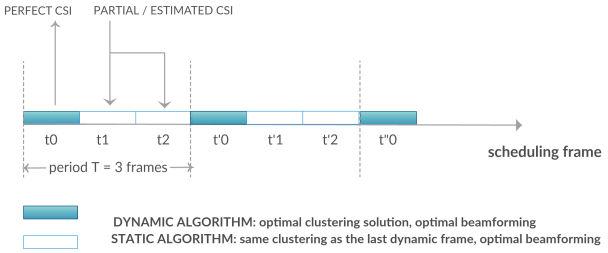


Fig. 2: Reference ABUC scheme with $T = 3$

- *Partial CSI knowledge:* perfect and full CSIs are only available at dynamic frames (every period T), and new updated CSIs are not available for static frames, i.e., they are assumed to be equal to the CSIs received in the previous dynamic frame (although the real channel states vary on a frame-by-frame basis, owing to random fading and user mobility).
- *Estimated CSI knowledge:* perfect and full CSIs are only available at dynamic frames (every period T), while for static frames, the CSI estimation model Eq. (3) will be applied.

Combining the periodicity T and the CSI feedback strategies results into different variants of the reference ABUC algorithm: parameters *(T, f_k)* means that dynamic clustering is performed every T frames and the type of CSI feedback $f_k \in \{f, p, e\}$ is applied, where f refers to Full CSI, p to Partial CSI and e to Estimated CSI, respectively.

Finally, the reference ABUC algorithm is described in Algorithm 3.

Algorithm 3: Reference ABUC Scheme with fixed parameters (T, f_k), $\forall k$

initialize frame t = 0, user velocity v
repeat
if $t \mod T = 0$ **then**
Get perfect CSI $\mathbf{H}_k(t)$ for all users k
Call Dynamic Clustering Algorithm
else
if *Full CSI* **then**
Get perfect CSI $\mathbf{H}_k(t)$ for all users k
else if *Partial CSI* **then**
Use imperfect CSI $\hat{\mathbf{H}}_k(t) = \mathbf{H}_k(t - \mod(t,T))$, for all users k
else
Estimate CSI $\hat{\mathbf{H}}_k(t)$ following (3) for all users k
Call Static Clustering Algorithm
Set clustering solution as the initial clusters for frame t+1
Move to next frame
until *convergence*

In [25], we showed that ABUC algorithm was able to balance the performance-cost trade-off under various mobility

profiles. However, the CSI feedback parameters, i.e., period T and CSI feedback type, were chosen empirically based on extensive simulations. To overcome this shortcoming, we proposed in [1] a Q-learning based algorithm to optimize its scheduling parameters on-the-fly, which is only applicable to a small network due to its high complexity and is unable to follow the rapid fluctuations of realistic wireless networks, in particular under the high mobility scenarios in consideration. Therefore, in this work, we design a novel and scalable framework by using a Deep Q-learning approach, presented in the sequel. Note that, instead of solving both the beamforming and user association optimization problem (P1) directly through DRL, we deliberately make use of mathematical optimization to solve (P1) and DRL to optimize the scheduling parameters of ABUC algorithm. This is because the beamforming solution of (P1) is very sensitive to CSI accuracy and hence mathematical optimization provides the best achievable solution, given (imperfect) CSI knowledge, which is available by default. That is, resolving problem (P1) through DRL would still require near-instantaneous CSI knowledge, obtained through an extremely fine granularity of CSI quantization. This would entail a prohibitively large state-space dimension, making the problem intractable. On the contrary, if the state-space dimension were limited to a tractable size through a coarser CSI quantization level, the performance of the DRL-based beamforming would suffer tremendous degradation. The rationale and originality of our proposed method is hence to ideally combine the advantages of those two approaches: mathematical optimization for fully exploiting the available CSI for beamforming and user association, and DRL for optimizing the meta-parameters of ABUC scheduling.

III. Basics on deep Q-learning for Partial observed Markov decision process (POMDP)

A. Background on reinforcement learning

Reinforcement learning is a major branch of machine learning, where an agent interacts with the environment in order to select optimal actions given its current state, in order to maximize its own reward function. The task of RL can usually be modeled as a Markov decision process (MDP), however, explicit transition probabilities and reward functions are not always available [26].

Q-learning is a basic reinforcement learning technique that does not require a model of the environment and that can handle problems with stochastic transitions and rewards. Each learning agent maximizes its accumulated future reward by adding the maximum value achieved from the next states to the reward of its current state, therefore successfully affecting the current action by the potential future reward. This potential function is a weighted sum of the expected values of all future steps beginning from the actual state.

In the Q-Learning (QL) algorithm, the agent decides to choose an action in each decision epoch and observes the results from this one. Each pair of action-state produces a Q-value that is updated in a Q-table in which the columns represent the possible actions and the rows describe states. The Q-value $Q^*(s_t, a_t)$ is updated by the Bellman function as follows [26]:

$$Q^*(s_t, a_t) = Q(s_t, a_t) + \alpha\big[\Gamma_t(s_t, a_t) + \eta \max_{s_{t+1}} Q^{t+1}(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\big], \tag{10}$$

where the Q-value $Q(s_t, a_t)$ is received when executing action a_t at state s_t and Γ_t is the system reward in time slot t. Parameters α and η are the learning rate and discount rate of the future expected reward, respectively. After updating the Q-table, in the next decision epoch, the agent is in a new state s_{t+1} and selects either the action corresponding to the higher Q-value $Q(s_{t+1}, a_{t+1})$ for exploitation, or a random action for exploration, as successful RL heavily relies on the exploitation-exploration trade-off. This algorithm is referred as the ϵ-greedy QL algorithm.

B. Background on POMDP

In our problem, each user feeds back its current CSI state, which consists of either full, partial or estimated instantaneous CSI values, as detailed in Section II-C3. These instantaneous values can be directly used to obtain the optimized beamforming and user association solutions through ABUC algorithm. However, to make use of DRL, the corresponding SINR levels need to be quantized in order to define a tractabe system state space with reasonable dimensions. Additionally, these SINR values at the DRL input are different from the real SINR states due to other effects: the feedback delay causing outdated CSI knowledge, and the feedback strategy of our ABUC algorithm which uses partial or estimated CSI (especially for $T_k > 1$).

By contrast to ideal MDP modeling, the full state of the system is often undetermined in realistic wireless networks as in our system. Therefore, we propose to use POMDP in place of conventional MDP for modeling our system, which better suits our targeted problem. POMDP models the lack of knowledge of the true underlying state by a probability distribution over the set of possible states, as introduced in [27]. Generally, a POMDP is described as a 6-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{O}, \Omega)$ where $\mathcal{S}, \mathcal{A}, \mathcal{P}$, and $\mathcal{R}$ are the states, actions, transitions, and rewards, defined similarly as for an MDP. The difference now is the introduction of the observation space $\mathcal{O}$ and its probability distribution Ω. The agent, instead of having the true system state s, receives an observation $o \in \mathcal{O}$ according to the probability distribution $o \sim \Omega(s)$.

Note that the considered process can be assumed Markovian as the next SINR state of each user - known at the cloud scheduler - only depends on the SINR state in the previous scheduling frame (owing to channel correlation), and the previous action of that user, namely, the CSI feedback period and scheme.

As we cannot observe the full SINR states in our problem, it should be modeled as a POMDP. Indeed, as represented in the SINR formula in section II, the value of SINR of user k, γ_k, is a function of channel $\mathbf{H}_k$. Hence, the quantized state of γ_k depends completely on the CSI feedback strategy chosen at each time slot.

IV. ABUC's Deep Q-learning framework

In this section, we detail the proposed DQL framework which enables to optimize the scheduling parameters of ABUC algorithm on-the-fly in the case of a network with a large number of users having different mobility profiles.

As we model our problem as a POMDP, we firstly define the system observation state, the action space and reward function for our DQL model.

1) System Observation State: The system observation state o_t is composed of the observed states of all K users. We partition and quantize the range of the perceived SINR γ_k into N levels. Each level corresponds to an observed state of the user SINR, i.e., each user k's observed state is defined as the quantized SINR level n_k^t, where $1 \leq n_k^t \leq N, n_k^t \in \mathbb{N}$, which is fed back to the cloud centralized scheduler so as to monitor the effect of the previous scheduling parameters' choices. Note that this feedback is additional to the underlying CSI feedback itself, which may be Full, Partial or Estimated, as explained in Section II-C. The system observed state at time slot t is hence defined as,

$$o_t = \{n_1^t, n_2^t, \ldots, n_k^t\}.$$

Note that the real system state $s_t = \{s_1^t, s_2^t, \ldots, s_k^t\}$ is given by the real SINR values experienced by each user, but unknown at the cloud centralized scheduler.

2) System Action: In the system, the central scheduler has to decide the feedback parameters to be selected. Let $\mathcal{T}$ and $\mathcal{F}$ denote the set of possible values of T and CSI feedback schemes, namely

$$\mathcal{T} = \{T_1, \ldots, T_p, \ldots, T_P\}$$
$$\mathcal{F} = \{f_1, \ldots, f_q, \ldots, f_Q\},$$

where $P \in \mathbb{N}$ and $Q \in \mathbb{N}$ represent the number of all possible values of period T and of types of CSI feedback schemes, respectively.

The current composite action a_t is denoted by

$$a_t = \{a_1^t, a_2^t, \ldots, a_k^t\}$$

where $a_k^t = (T_k^t, f_k^t)$ represents the feedback parameters of user k at time slot t, where period $T_k^t \in \mathcal{T}$ and CSI feedback type $f_k^t \in \mathcal{F}$.

3) Reward Function: The system reward needs to represent the optimization objective, that is to simultaneously reduce the system cost and satisfy the sum-rate demands. Here, we define the overall system reward at observed state o_t and action a_t as

$$\Gamma_t(o_t, a_t) = \rho_1 \sum_{k=1}^{K} r_k(o_k^t, a_k^t) - \rho_2 \sum_{k=1}^{K} C_k(o_k^t, a_k^t), \quad (11)$$

where the first term is the achieved system sum-rate, where the rate r_k of user k at the observed state o_k^t is given by Eq. (2) in which the beamforming vector w_k has been optimized based on the CSI feedback according to previous action a_k^{t-1}. The second term denotes the CSI signaling overhead induced by action a_k^t, at observed state o_k^t. Weighting parameters $\rho_1, \rho_2 \in [0, 1]^2$ represent the trade-off between sum-rate and cost, where $\rho_1 + \rho_2 = 1$.

The CSI overhead cost $C_k(o_k^t, a_k^t = \{T_k, f_k\})$ of each user k is computed over T_k frames and can be expressed as follows [25]:

- If Full CSI, $f_k = f$:

$$C_k(o_k^t, a_k^t = \{T_k, f\}) = \frac{1}{T_k}\left[\big[L + (T_k - 1)L_k\big]MN\right], \quad (12)$$

- If Partial $f_k = p$ or Estimated CSI $f_k = e$:

$$C_k(o_k^t, a_k^t = \{T_k, p\}) = C_k(o_k^t, a_k^t = \{T_k, e\}) = \frac{LMN}{T_k}. \quad (13)$$

Deep Q-learning algorithm uses an ϵ-greedy strategy [28] in which the amount of exploration is controlled by the parameter ϵ. The agent selects a random action with a given probability ϵ, $0 \leq \epsilon \leq 1$. At first, this rate must be initialized to a sufficiently high value, i.e, $\epsilon = 1$, and then be decayed progressively after getting more knowledge about the environment.

We also define a super-frame composed of F_0 successive scheduling frames in which the same action is executed. The obtained reward for the corresponding action is averaged over every single frame in a super-frame.

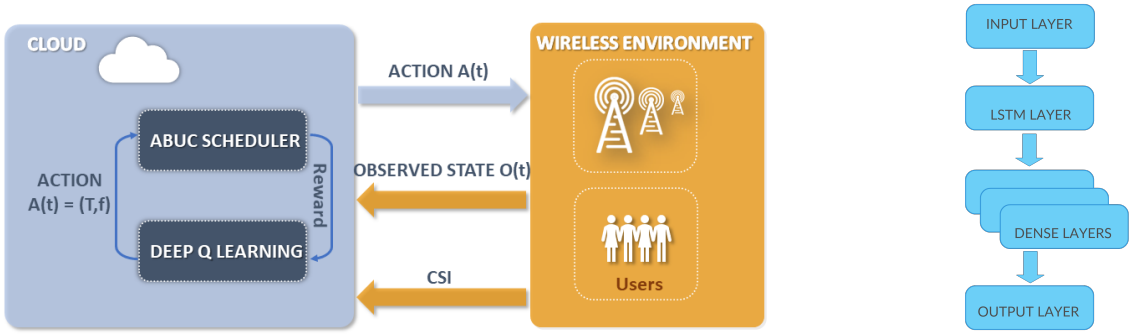
Different to Q-learning approach, the agent does not have the global knowledge about the expected reward value for each state-action pair, but it is learned by experience over subsets of state-action pairs. A method that trains the neural network with experiences in the memory is called Experience Replay. In this method, each experience (consisting of the current state, action, reward, and next state) obtained by the agent is stored in the experience replay memory. Instead of training the neural network based on the agents' actual observations, past experiences are sampled from this replay memory by means of the minibatch method.

By using the *replay*, the experiences used to train the Deep Q-network (DQN) come from many different points in time, thereby smoothing out learning and avoiding severe failures.

In addition, to maintain the experiences' history, we make use of a memory-based layer in the neural network, namely the Long short-term memory (LSTM) [27]. LSTM method appears to be well-suited as it manages to keep the contextual information of the neural network's inputs allowing information to flow from one step to the next, thereby resulting into learning improvement of the neural network.

The general framework of our proposed ABUC's DQL is presented in Fig. 3a. From the wireless environment, the system state and the user CSI are fed back to the cloud where ABUC scheduler and Deep Q-learning network are implemented. ABUC scheduler takes as inputs the action given by the DQN in each decision phase and solves the optimal beamforming and user clustering. This allows to calculate the new reward that is returned to the DQN. The decision is also fed back to the wireless environment such that each user can update its feedback method for the next frames.

In Fig. 3b, we present the structure of the Deep Neural Network used in our framework. The input and output layers have the size of observed state space and action space, respectively. The LSTM layers are inserted after the input layer and a number of hidden dense layers improve the computational capabilities of the Neural network.



(a) ABUC's DQL framework (b) DNN design

Fig. 3: Proposed Deep Q-learning framework

The details of the proposed ABUC's Deep Q-learning framework are given in Algorithm 4. In the first super-frame $i = 1$, we initialize the exploration rate by $\epsilon \leftarrow 1$. Then, during each super-frame, we generate a random probability η and compare its value to ϵ to decide how an action is selected. Once a decision is made, we execute the ABUC algorithm (as shown in Algorithm 1) during F_0 internal frames and compute the necessary parameters such as sum-rate, CSI cost and reward according to (11). At the end of each decision, the transition tuple (o_i, a_i, o_{i+1}) is saved such that we can do the *replay* when the memory has sufficient elements for training. Finally, we reduce progressively ϵ and exploration decay ξ.

The complexity of our DQL algorithm is function of the state and action space sizes which can be large for a high number of users. However, it is known that the DQL approach based on DQNs is more efficient to deal with large state action spaces and highly dynamic environments [21]. As to the intrinsic complexity of ABUC algorithm, the reader may refer to our previous works in [24], [25]. Under our architectural assumptions, the complexity burden induced by the DQL-based ABUC algorithm can reasonably be handled by the large cloud processing capacity.

V. Simulation Results

In this section, we present the performance results of the proposed ABUC's Deep Q-learning framework in a H-CRAN network. We consider a two-tier H-CRAN which consists of a single macro-RRH and 3 pico-RRHs equally separated in space. Up to 12 mobile users are uniformly distributed over the macro cell. We assume a Random Waypoint model to represent users' movements. The fronthaul constraints for macro-RRH and pico-RRH are 683.1 Mbps and 106.5 Mbps, respectively [13]. All channels undergo Rayleigh small scale fading and log-normal shadowing. The other parameter settings can be found in Table I.

The wireless environment and ABUC algorithm are implemented in Matlab, while Deep Q-Network program is developed in Python. The neural network and LSTM layer are based on Keras. The convergence of an action is determined if it is maintained during at least 30 consecutive super-frames. The number of single frames in each super-frame is set to 10.

Algorithm 4: Proposed ABUC's Deep Q-learning framework

initialization initialize replay memory
F_{max}: number of super-frames
F_0: number of frames for each super-frame
ϵ: exploration rate, ϵ 1
ξ : exploration decay
$minibatch$: number of randomly sampled elements of the memory for replay
for *super-frame i = 1:* F_{max} **do**
  if *i = 1* **then**
    With probability ϵ, randomly select an action
  else
    Randomly generate a probability ξ
    if $\xi \leq \epsilon$ **then**
      randomly select an action
    else
      choose action $a_i = arg\max Q(o_i, a_i)$
    end
  end
  for *frame t=1:*F_0 **do**
    Execute ABUC with a_i^k *parameter*
    Obtain beamforming and clustering solutions for each frame t
    Compute average sum-rate of super-frame i over all F_0 *frames*
  end
  Compute the reward Γ_i *using* (11) *and observe the new state* $o_{i+1} = \{n_i^k\}$
  Store transition (o_i, a_i, o_{i+1}) *in replay memory.*
  Get $minibatch$ *samples from memory for training the neural network*
  Call replay function
  Reduce exploration rate $\epsilon = \epsilon \times \xi$
end

Simulation parameters	
Cellular layout	Hexagonal two-tier H-CRAN
Channel bandwidth	10MHz
Intercell distance	0.8km
TX power for macro/pico RRH	(43, 30) dBm
Antenna gain	15 dBi
Background noise	-169 dBm/Hz
Path-loss from macro RRH to user	128.1 + 37.6 log10(d)
Path-loss from pico RRH to user	140.7 + 36.7 log10(d)
Log-normal shadowing	8 dB
CSI error variance	-20 dB
Scheduling frame	10 ms
User priority weights α_k	1 $\forall k$

TABLE I: Parameter settings for simulations

To evaluate the algorithm's behavior with regard to user mobility, we consider different mobility profiles represented by the parameter λ_v which is a function of velocity. We set the carrier frequency f_c and the fronthaul delay T_{dl} for all RRHs l as 900 MHz and 2 ms [29], respectively.

The agent in the cloud will learn over super-frames for an observed state space of 6^K states, corresponding to the 6 quantized SINR levels of each user. The action space for each user consists of 7 actions, giving the action space size of 7^K: $T = 1$ with Full CSI $(1, f)$; $T = 2$ with Full CSI $(2, f)$; $T = 3$ with Full CSI $(3, f)$; $T = 2$ with Estimated CSI $(2, e)$; $T = 3$ with Estimated CSI $(3, e)$; $T = 2$ with Partial CSI.

We consider two scenarios: homogeneous and heterogeneous scenarios in terms of user velocity. In the first one, all users have the same velocity while in the second, each user has its own individual velocity.

A. DQL performance in homogeneous mobility scenario

Firstly, we evaluate the proposed algorithm in a H-CRAN network where 9 users undergo the same velocity which is varied over the three mobility profiles, namely: low, medium and high mobilities corresponding respectively to 5 km/h, 40 km/h and 80 km/h. We vary the value of weights (ρ_1, ρ_2) in five cases: (0.1, 0.9) (0.3, 0.7), (0.5, 0.5), (0.7, 0.3) and (0.9, 0.1) representing different trade-offs between sum-rate (expressed in Mbit/s) and CSI feedback costs.

Fig. 4 shows the convergence behavior of the average system reward over super-frames for each weight pair. First of all, we can see that the system reward increases with the weight ρ_1 as the value of the sum-rate is dominant over the cost. We observe that when ρ_1 approaches 0, e.g, $\rho_1 = 0.1$ in Fig. 4a, the sum-rate factor is much less prevalent than that of CSI cost, hence users choose the action that minimizes their cost by using the maximum period value of $T = 3$ and avoiding Full CSI feedback. In this case, as the channel estimation has good quality for low and medium mobility, i.e., $(3, e)$ is chosen as optimal action. While in case of high mobility, $(3, p)$ is used instead because CSI estimation worsens. Inversely, when ρ_1 approaches 1, e.g, $\rho_1 = 0.9$ in Fig. 4e, the reward tends towards the sum-rate, hence all users actions with any velocity converge to Full CSI feedback with $T = 1$ as it provides the best sum-rate performance.

For the other values of ρ_1, the converged actions provide an optimized trade-off between sum-rate and CSI feedback cost.

Mobility	$\rho_1 = 0.1$	$\rho_1 = 0.3$	$\rho_1 = 0.5$	$\rho_1 = 0.7$	$\rho_1 = 0.9$
Low	$(3, e)$	$(3, e)$	$(3, e)$	$(2, e)$	$(1, f)$
Medium	$(3, e)$	$(3, e)$	$(3, f)$	$(2, f)$	$(1, f)$
High	$(3, p)$	$(3, p)$	$(3, f)$	$(2, f)$	$(1, f)$

TABLE II: Synopsis of optimal actions

In Fig. 4b, although the users maintain the same converged actions as in Fig. 4a as ρ_1 is still low, the reward gap between low mobility and the two others mobility profiles is enlarged. We can observe the biggest difference between the reward of low mobility compared to medium and high mobility in Fig. 4c, which is mainly due to the actions adopted by the users in this case. When low mobility users still use Estimated CSI, the rest have to apply Full CSI to avoid the sum-rate degradation, but suffer a much higher CSI overhead. In Fig. 4d, as the sum-rate begins to overcome CSI cost in weight, i.e, as $(\rho_1, \rho_2) = (0.7, 0.3)$, all users tend to guarantee a better sum-rate by reducing their period to $T = 2$.

To conclude, regarding the action and the reward, we can observe that low velocity users can afford to perform CSI estimation rather than using partial or full knowledge of CSI (except for $\rho_1 = 0.9$). The reason is that the channel quality is expected to be more stable in low velocity case and the CSI estimation performs with accurate precision according to Eq. (3). By contrast, in case of higher velocity, the CSI estimation is no longer accurate and the sum-rate performance degrades significantly. Then, for a given mobility profile, when the sum-rate weight ρ_1 increases, users tend to switch from Estimated and Partial CSI to Full CSI to get a better sum-rate, and to reduce their period T to limit performance loss in static frames. The optimal actions are summarized in Table II.

B. DQL performance in heterogeneous mobility scenario

In this scenario, we also consider the same H-CRAN network including 9 users displaying three different mobility profiles. Users 1-2-3 have low velocity, users 4-5-6 have medium velocity and the rest of users have high velocity. The sum-rate coefficient ρ_1 is varied over three values: 0.1, 0.5 and 0.9.

In Fig. 5, we present the convergence of individual reward, rate and CSI cost for $\rho_1 = 0.1$. As for the homogeneous scenario, we can see that the users converge to the same optimal action based on their own individual profile. The group of low mobility users always obtains the highest values of reward thanks to the most accurate and stable quality of CSI feedback. Meanwhile, the groups of higher velocity often suffer more degradation in terms of sum-rate and also pay higher CSI cost to get more accurate information from Partial CSI and Full CSI feedback.

The other results of converged reward, rate and action of individual user for $\rho_1 = 0.5$ and $\rho_1 = 0.9$ are given in Tables III and IV. Again, we observe the same tendency in terms of converged action as in homogeneous scenario.

According to the logged positions derived during the experiments, as the low mobility users did not move much, we can observe in Fig. 5b that the initial position may somewhat have influence on the individual rate performance. For example, user 2 has clearly a better rate than that of users 1 and 3

User	1	2	3	4	5	6	7	8	9
Reward	7.5	7.6	6.8	5.4	4.8	5.3	5.1	5.0	5.7
Sum-rate	21.0	22.3	21.2	22.2	21.8	21.5	22.0	21.1	22.9
Action	$(3, e)$	$(3, e)$	$(3, e)$	$(3, f)$	$(3, f)$	$(3, f)$	$(3, f)$	$(3, f)$	$(3, f)$

TABLE III: Converged individual reward, rate and action per user for $\rho_1 = 0.5$

User	1	2	3	4	5	6	7	8	9
Reward	21.7	22.3	21.0	20.8	20.2	20.0	21.0	20.5	22.9
Sum-rate	26.4	27.1	25.6	25.4	24.7	24.5	25.6	25.0	27.7
Action	$(1, f)$	$(1, f)$	$(1, f)$	$(1, f)$	$(1, f)$	$(1, f)$	$(1, f)$	$(1, f)$	$(1, f)$

TABLE IV: Converged individual reward, rate and action per user for $\rho_1 = 0.9$

thanks to a more favorable position over the scheduling time, as users with the best channel conditions are more likely to be served more often by the sum-rate maximization scheduler. In Fig. 5c, all users converge to the same CSI cost value as both actions $(3, e)$ and $(3, p)$ induce the same cost.

In Fig. 6, we present the convergence of the individual CSI overhead cost obtained for two other values of ρ_1. In Fig 6a, the users from 4 to 9 who adopt action $(3, f)$ naturally converge all to a higher cost value than that of low mobility users, as action $(3, e)$ generates less CSI feedback owing to the CSI estimation strategy. In Fig 6b, once again all users converge to same value of CSI cost as all of them undertake the same action $(1, f)$ at convergence.

C. Optimal action and convergence time against different number of users

To examine whether and how the number of users may influence the converged action and the convergence time, we carried this set of experiments by varying the number of users from 6, 9 to 12 users in a heterogeneous mobility case with $\rho_1 = 0.1$. The results are presented in Fig. 7.

Table V summarizes additional results concerning the convergence time, optimal action and total user rate per mobility profile as function of the number of users. We can see from Table V and Fig. 5 that the convergence time increases with the number of users, as the observed state space and action space grow exponentially as a function of the number of users K. Hence, the algorithm takes more time to explore all possible actions to finally converge to the optimal ones. However, it is important to notice that whatever the value of K, the optimal actions remain the same for each mobility profile.

D. Distribution of selected actions

In order to analyze how each action is picked during the whole scheduling process until convergence, we plot in Fig. 8 the distribution of the selected actions before convergence is reached, for each type of user mobility.

In these figures, we can observe that the optimal action in each case is always the most selected. In addition, looking at the most frequently selected action, we can notice that the choices are coherent with the sum-rate-CSI trade-off, i.e., the values of (ρ_1, ρ_2). For example, when observing the results for $\rho_1 = 0.1$ (blue bars), the two best actions are always $(3, e)$ and $(3, p)$ regardless of the velocity. This is because at the lowest value of ρ_1, the users prefer the largest period $T = 3$ with Estimated CSI and Partial CSI. Therefore, every option with Full CSI, i.e., $\{1, 2, 3\}$ or $\{(1, f), (2, f), (3, f)\}$ must be quickly eliminated from the candidate action set.

In Fig. 8a, for $\rho_1 = 0.5$ (green bars), the most frequent actions for low mobility users are still that of Estimated CSI and Partial CSI, but in this case, the two best ones are $(2, e)$ and $(3, e)$ because they offer the better sum-rate at low velocity compared to Partial CSI. For the same weight in Fig. 8b, $(2, f)$ and $(3, f)$ are the most selected actions since the estimation of CSI becomes less accurate.

For $\rho_1 = 0.9$ (yellow bars), as the sum-rate almost dominates over the total reward function, $(1, f)$ is always the best choice for all mobility profiles. Somehow, we can see that Estimated CSI is still competitive in Fig. 8a at low mobility thanks to a good estimation quality. This action is much less selected in case of higher velocities as shown in Fig. 8b.

The convergence times of individual users are summarized in Table VI. We observe almost similar convergence time intervals that are around 5000-6000 super-frames even with different initial positions and user velocities. Such observation consolidates the conclusion that the convergence time is mainly impacted by the number of users K rather than by the individual characteristics of users.

E. Adaptability of the online-learning in dynamic environments

In real networks, the user mobility can vary in real-time and the current agent's knowledge about the environment cannot be applicable anymore in solving the optimal action. Therefore, it is necessary for the system to adapt to the environment's changes and rapidly re-learn the optimal policy. In Fig. 9, we present the short moving average of total reward in a network of 3 users having different velocities (5,40,80) km/h at the beginning of the experiment. The current scenario setting achieves a convergence after 443 super-frames with optimal actions $(3e, 3e, 3p)$. At super-frame 800, user 1 who has low mobility (5 km/h) switches its velocity to 80 km/h. This sudden change makes an important degradation on the total reward because the CSI estimation quality is deteriorated as user 1 is selecting an action ($T = 3$, e) which is no more appropriate to the real experienced channels.

At super-frame 851, the agent detects the change of mobility by recognizing a drop of 5% of the moving average of converged reward (phase 1). At this point, the exploration rate ϵ is reset to allow the agent start learning the new behavior of users. As we can see in the figure, the new convergence

Mobility	6			9			12		
	CV	action	SR	CV	action	SR	CV	action	SR
Low	2958	$(3, e)$	46.96	5862	$(3, e)$	63.83	11866	$(3, e)$	101.48
Medium	2901	$(3, e)$	34.09	6198	$(3, e)$	47.44	12152	$(3, e)$	72.97
High	2954	$(3, p)$	21.83	6264	$(3, p)$	37.81	11625	$(3, p)$	44.17

TABLE V: Convergence per mobility profile, CV = convergence time, SR = sum-rate

User	1	2	3	4	5	6	7	8	9
$\rho_1 = 0.1$	5860	5862	4342	5188	5664	6198	5190	5334	6264
$\rho_1 = 0.5$	5837	5244	6004	5837	6018	5658	5922	5905	5242
$\rho_1 = 0.9$	5497	4884	5634	5982	5837	5188	5922	5982	5712

TABLE VI: Synopsis of convergence time (expressed in number of super-frames) in case of heterogeneous mobility users

is established at super-frame 1225 with new optimal actions $(3p, 3e, 3p)$, that better adapt to the individual change of user 1.

This result shows a good behavior of the online-learning of our framework to adapt in real-time network operation. The time of adaptation can be adjusted by acting on the parameter of detection (e.g., reward drop $\leq 5\%$) for a better reactivity.

VI. Conclusion and Perspectives

In this paper, we designed a deep reinforcement learning framework which enables our previously proposed ABUC algorithm to optimize its scheduling parameters on-the-fly, given each user mobility profile. We proposed a deep Q-learning algorithm based on an POMDP model to better tackle the scalability issue of our targeted problem.

More specifically, we have shown that our proposed DQL framework can achieve attractive results in terms of converged action and obtained reward in both homogeneous and heterogeneous mobility scenarios. The experiments prove that the convergence time is mainly impacted by the number of users in the network. They also demonstrated the online-learning ability of the framework to rapidly adapt to the changes of users mobility.

As a future work direction, it would be interesting to enhance the proposed framework by further investigating the users fairness parameter α_k. Another interesting direction would be to couple this parameter with the heterogeneous users QoS requirements in view of the upcoming Beyond 5G and 6G applications. A particular attention should be devoted to emerging technologies such as RIS and holographic MIMO surfaces [30] whose integration would further improve the performance of heterogeneous networks. Finally, as energy efficiency will become a major key performance indicator in future 6G networks, it is essential to adapt our proposed methods to handle such energy-related issues [31], [32].

Acknowledgement

This work was supported by the CNRS PICS bilateral research fund between France and Japan, and by the Grants-in-Aid for Scientific Research (Kakenhi) no. 17K06453 and no. 20H00592 from the Ministry of Education, Science, Sports and Culture of Japan.

References

[1] D. T. Ha, L. Boukhatem, M. Kaneko, N. Nguyen-Thanh, and S. Martin, "Adaptive beamforming and user association in heterogeneous cloud radio access networks: A mobility-aware performance-cost trade-off," *Computer Networks*, vol. 160, pp. 130 – 143, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S138912861830714X

[2] A. Ijaz, L. Zhang, M. Grau, A. Mohamed, S. Vural, A. U. Quddus, M. A. Imran, C. H. Foh, and R. Tafazolli, "Enabling massive IoT in 5G and beyond systems: Phy radio frame design considerations," *IEEE Access*, vol. 4, pp. 3322–3339, 2016.

[3] M. Peng, Y. Li, Z. Zhao, and C. Wang, "System architecture and key technologies for 5G heterogeneous cloud radio access networks," *IEEE Network*, vol. 29, no. 2, pp. 6–14, March 2015.

[4] N. Chen, B. Rong, X. Zhang, and M. Kadoch, "Scalable and flexible massive MIMO precoding for 5G H-CRAN," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 46–52, February 2017.

[5] G. C. Alexandropoulos, P. Ferrand, J. Gorce, and C. B. Papadias, "Advanced coordinated beamforming for the downlink of future LTE cellular networks," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 54–60, 2016.

[6] Q. Shi, M. Razaviyayn, Z. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.

[7] G. C. Alexandropoulos, P. Ferrand, and C. B. Papadias, "On the robustness of coordinated beamforming to uncoordinated interference and CSI uncertainty," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6.

[8] M. Peng, Y. Li, J. Jiang, J. Li, and C. Wang, "Heterogeneous cloud radio access networks: a new perspective for enhancing spectral and energy efficiencies," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 126–135, December 2014.

[9] H. Dahrouj, A. Douik, O. Dhifallah, T. Y. Al-Naffouri, and M. S. Alouini, "Resource allocation in heterogeneous cloud radio access networks: advances and challenges," *IEEE Wireless Communications*, vol. 22, no. 3, pp. 66–73, June 2015.

[10] X. Mao, B. Zhang, Y. Chen, J. Yu, and Z. Han, "Matching game based resource allocation for 5G H-CRAN networks with device-to-device communication," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct 2017, pp. 1–6.

[11] M. M. U. Rahman, H. Ghauch, S. Imtiaz, and J. Gross, "RRH clustering and transmit precoding for interference-limited 5G CRAN downlink," in *2015 IEEE Globecom Workshops (GC Wkshps)*, Dec 2015, pp. 1–7.

[12] B. Dai and W. Yu, "Sparse beamforming and user-centric clustering for downlink cloud radio access network," *IEEE Access*, vol. 2, pp. 1326–1339, 2014.

[13] ——, "Backhaul-aware multicell beamforming for downlink cloud radio access network," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, June 2015, pp. 2689–2694.

[14] M. Peng, Y. Yu, H. Xiang, and H. V. Poor, "Energy-efficient resource allocation optimization for multimedia heterogeneous cloud radio access networks," *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 879–892, May 2016.

[15] C. Huang, R. Mo, and C. Yuen, "Reconfigurable intelligent surface assisted multiuser MISO systems exploiting deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, pp. 1839–1850, Aug. 2020.

[16] C. Huang, Z. Yang, G. Alexandropoulos, K. Xiong, L. Wei, C. Yuen, and Z. Zhang, "Hybrid beamforming for RIS-Empowered multi-hop Terahertz communications: A DRL-based method," in *2020 IEEE Globecom Workshops*, Dec 2020, pp. 222–226.

[17] Y. He, Z. Zhang, F. R. Yu, N. Zhao, H. Yin, V. C. M. Leung, and Y. Zhang, "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 433–10 445, Nov 2017.

[18] Y. He, N. Zhao, and H. Yin, "Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 44–55, Jan 2018.

[19] Y. Sun, M. Peng, and S. Mao, "Deep reinforcement learning based mode selection and resource management for green fog radio access networks," *IEEE Internet of Things Journal*, pp. 1–1, 2019.

[20] Z. Xu, Y. Wang, J. Tang, J. Wang, and M. C. Gursoy, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[21] 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.

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14236

[23] D. S. Michalopoulos, H. A. Suraweera, G. K. Karagiannidis, and R. Schober, "Amplify-and-forward relay selection with outdated channel estimates," *IEEE Transactions on Communications*, vol. 60, no. 5, pp. 1278–1290, May 2012.

[24] H. D. Thang, L. Boukhatem, M. Kaneko, and S. Martin, "Performance-cost trade-off of joint beamforming and user clustering in cloud radio access networks," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct 2017, pp. 1–5.

[25] D. T. Ha, L. Boukhatem, M. Kaneko, and S. Martin, "An advanced mobility-aware algorithm for joint beamforming and clustering in heterogeneous cloud radio access network," in *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWIM '18. New York, NY, USA: ACM, 2018, pp. 199–206. [Online]. Available: http://doi.acm.org/10.1145/3242102.3242120

[26] V. Francois-Lavet, P. Henderson, I. Riashat, M. G. Bellemare, and J. Pineau, *An Introduction to Deep Reinforcement Learning*. Now Publishers, 2019.

[27] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable MDPs," in *AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents (AAAI-SDMIA15)*, Arlington, Virginia, USA, November.

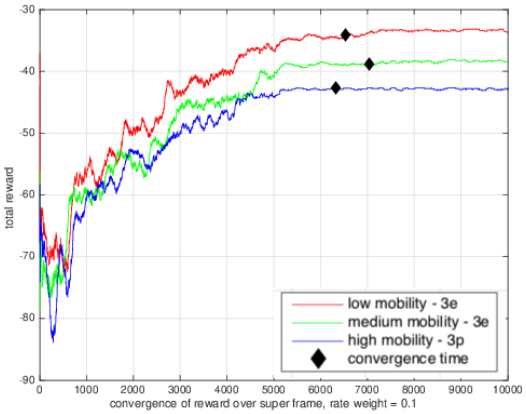
[28] M. Rovcanin, E. D. Poorter, I. Moerman, and P. Demeester, "A reinforcement learning based solution for cognitive network cooperation between co-located, heterogeneous wireless sensor networks," *Ad Hoc Networks*, vol. 17, pp. 98–113, 2014.

[29] 3GPP, "Technical specification group services and system aspects, quality of service (QoS) concept and architecture (release 12)," *Report TS 23.107, V12.0.0*, Sep 2014.

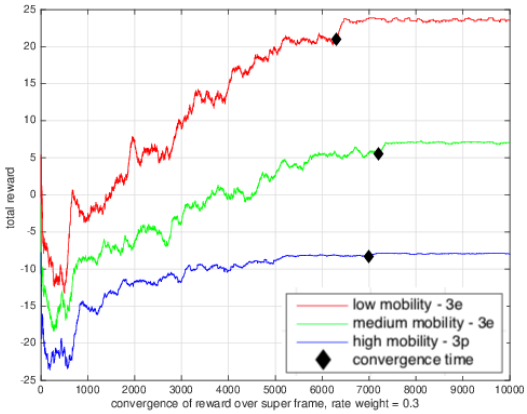
[30] C. Huang, S. Hu, G. Alexandropoulos, A. Zappone, C. Yuen, R. Zhang, M. Di Renzo, and M. Debbah, "Holographic MIMO surfaces for 6G wireless networks: Opportunities, challenges, and trends," *IEEE Wireless Communications*, vol. 27, pp. 118 – 125, Oct. 2020.

[31] C. Huang, A. Zappone, G. Alexandropoulos, M. Debbah, and C. Yuen, "Reconfigurable intelligent surfaces for energy efficiency in wireless communication," *IEEE Transactions on Wireless Communications*, pp. 4157 – 4170, Aug. 2019.

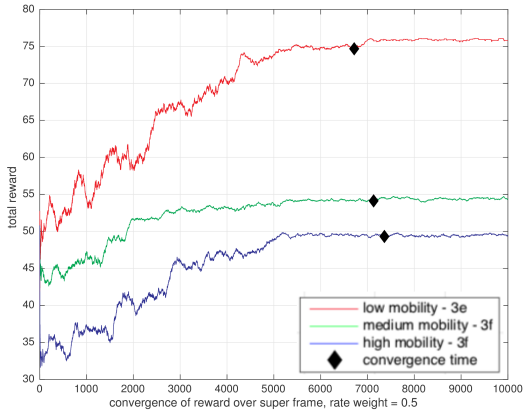
[32] T. Dinh, M. Kaneko, E. Fukuda, and L. Boukhatem, "Energy efficient resource allocation optimization in fog radio access networks with outdated channel knowledge," *IEEE Transactions on Green Communications and Networking*, vol. 5, pp. 146–159, March 2021.



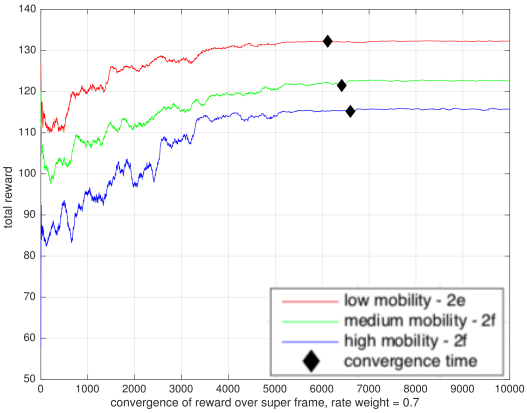
(a) $(\rho_1, \rho_2) = (0.1, 0.9)$



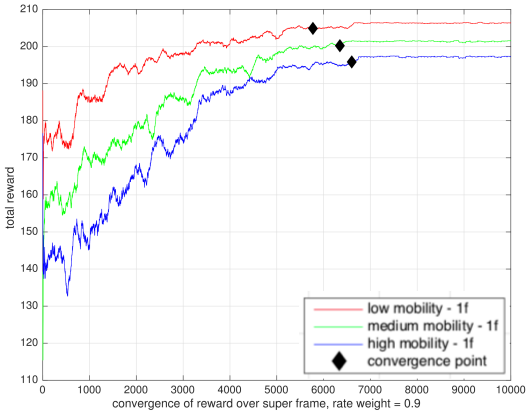
(b) $(\rho_1, \rho_2) = (0.3, 0.7)$



(c) $(\rho_1, \rho_2) = (0.5, 0.5)$

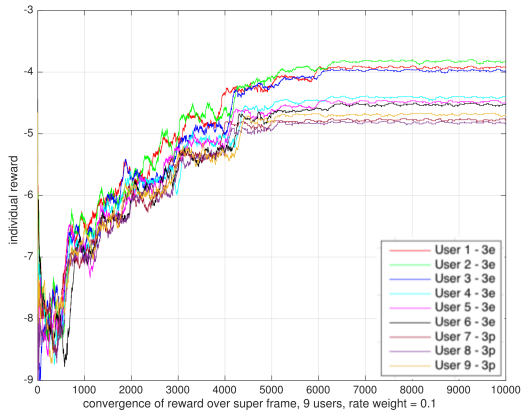


(d) $(\rho_1, \rho_2) = (0.7, 0.3)$

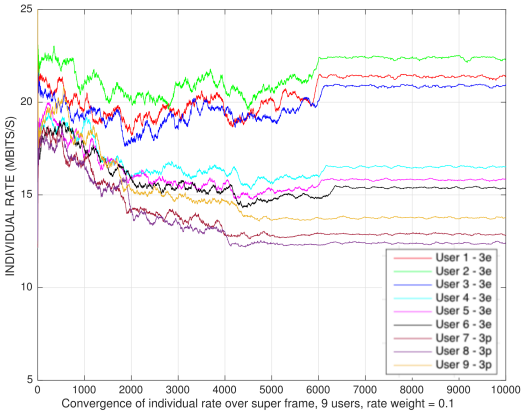


(e) $(\rho_1, \rho_2) = (0.9, 0.1)$

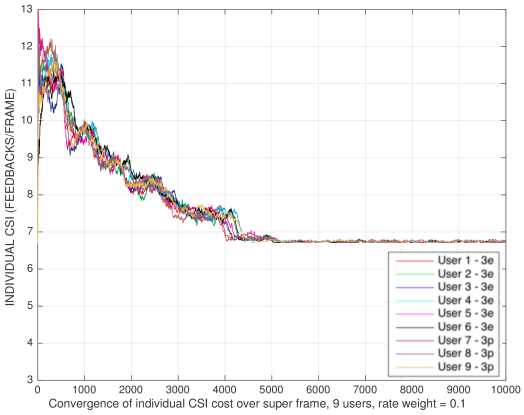
Fig. 4: Total reward convergence in homogeneous mobility scenario



(a) Individual reward

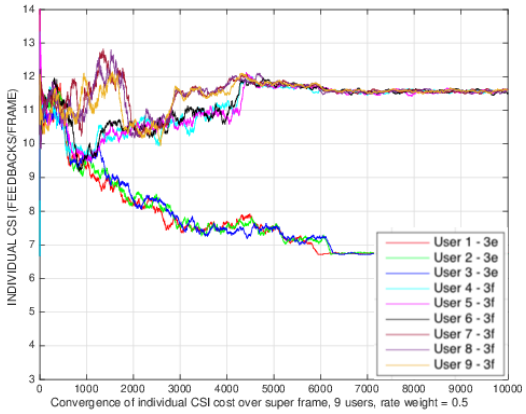


(b) Individual rate (Mbit/s)

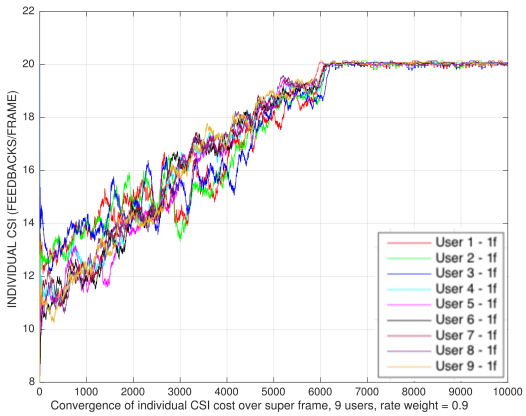


(c) Individual CSI cost (feedback/frame)

Fig. 5: Individual convergence of reward, rate and CSI cost in heterogeneous mobility scenario, $\rho_1 = 0.1$

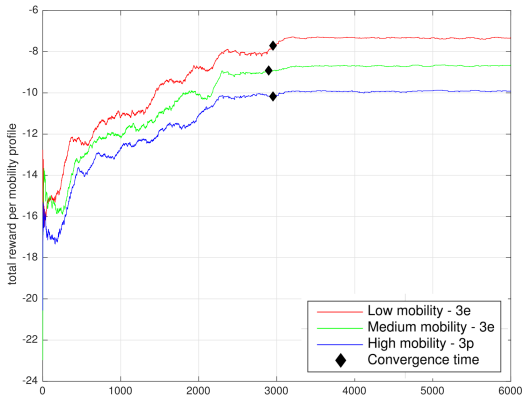


(a) $(\rho_1, \rho_2) = (0.5, 0.5)$

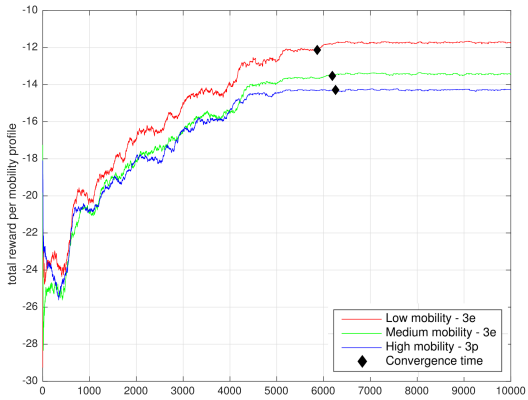


(b) $(\rho_1, \rho_2) = (0.9, 0.1)$

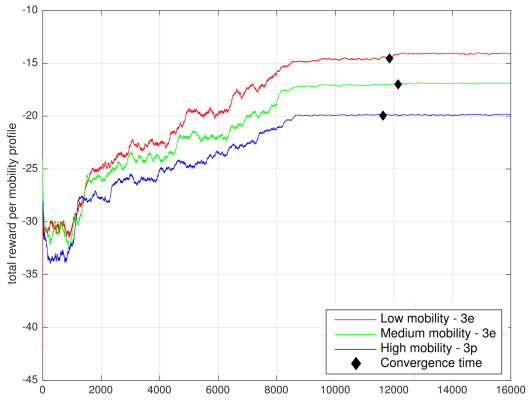
Fig. 6: Individual CSI cost convergence in heterogeneous mobility scenario



(a) 6 users

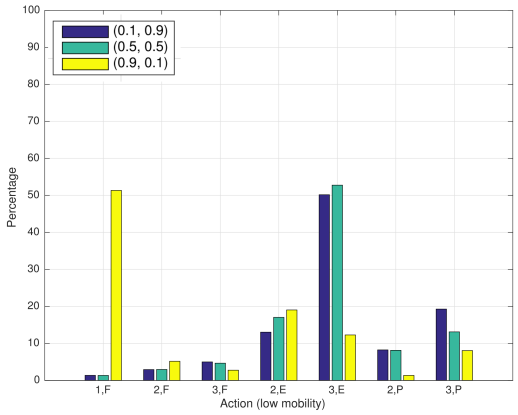


(b) 9 users

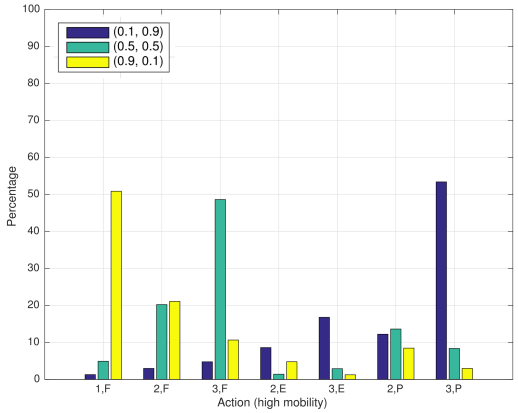


(c) 12 users

Fig. 7: Reward convergence per mobility profile in heterogeneous mobility scenario



(a) Low mobility users 1-3



(b) High mobility users 7-9

Fig. 8: Distribution of selected actions for different mobility profiles

Fig. 9: Reward convergence in case of online-learning