Collaborative Artificial Intelligence (AI) for User-Cell Association in Ultra-Dense Cellular Systems

Kenza Hamidouche, Ali Taleb Zadeh Kasgari, Walid Saad, Mehdi Bennis, Merouane Debbah

Collaborative Artificial Intelligence (AI) for User-Cell Association in Ultra-Dense Cellular Systems

Kenza Hamidouche*, Ali Taleb Zadeh Kasgari†, Walid Saad†, Mehdi Bennis‡ and Mérouane Debbah*§
* LSS, CentraleSupelec, Université Paris-Saclay, Gif-sur-Yvette, France Email: kenza.hamidouche@centralesupelec.fr.
† Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA Emails:{alitk,walids}@vt.edu.
‡ CWC - Centre for Wireless Communications, Oulu, Finland, Email: bennis@ee.oulu.fi
§ Mathematical and Algorithmic Sciences Lab, Huawei France R&D, Paris, France Email: merouane.debbah@huawei.com.

***Abstract*—In this paper, the problem of cell association between small base stations (SBSs) and users in dense wireless networks is studied using artificial intelligence (AI) techniques. The problem is formulated as a mean-field game in which the users' goal is to maximize their data rate by exploiting local data and the data available at neighboring users via an imitation process. Such a collaborative learning process prevents the users from exchanging their data directly via the cellular network's limited backhaul links and, thus, allows them to improve their cell association policy collaboratively with minimum computing. To solve this problem, a neural Q-learning learning algorithm is proposed that enables the users to predict their reward function using a neural network whose input is the SBSs selected by neighboring users and the local data of the considered user. Simulation results show that the proposed imitation-based mechanism for cell association converges faster to the optimal solution, compared with conventional cell association mechanisms without imitation.**

I. Introduction

The emergence of the Internet of Things (IoT) has given rise to a significant amount of data, collected from sensors, user devices, and base stations, that must be processed by next-generation wireless systems [1]–[5]. Relying on traditional cloud-centric approaches for big data analytics may no longer be suitable for dense cellular systems that encompass both IoT devices and conventional mobile phones. Instead, it has become imperative to leverage the distributed storage and computing power available in the network infrastructure and devices (e.g., smartphones, computers, tablets and base stations) so as to process the data. The data that is gathered at the devices and base stations (BSs) is primarily related to the network operations and includes the number of connections from a given device to a BS, the type of requested data, the traffic load at specific time periods, the location of the users, and the channel state information, among others. Clearly, such type of data is private in nature and users or BSs that are owned by different network operators would be reculant to share their own collected data.

Recently, machine learning-based artificial intelligence (AI) techniques [10], [17] have emerged as promising tools that allow a cellular network to leverage the aforementioned and optimize its various cross-layer functions. In particular, AI techniques provide distributed, self-organizing solutions to complex wireless networking problems such as resource allocation, decoding/encoding, and cell association [6], [13]. Indeed, the association of users to BSs in ultra-dense and time-varying cellular networking environments becomes challengingf to model and solve mathematically while capturing all the network dynamics. This motivates the need for addressing cell association problems using distributed learning algorithms that enable both users and BSs to exploit the data that can be gathered by BSs in the network.

The privacy constraints in cellular networks coupled with the traffic load induced by centralized learning frameworks makes it necessary to develop distributed machine learning algorithms for cell association [13]. These algorithms must be able to exploit the training data that is stored at a large number of devices to reduce the local training time and save their computing and spectrum resources. The main objective when designing a distributed learning framework is to allow a given user to benefit from the learning and processing of other neighboring users that have already selected their serving BS. For instance, users that are located in the same area will often experience the same channel condition, and the same distance to the BS separates them. Thus, a given user can exploit information about selected BSs by users that have similar network conditions.

The problem of cell association was extensively addressed in the literature [7]–[9] and [11]. The work in [8] addressed the problem of cell association between unmanned aerial vehicles (UAVs) and users using optimal transport theory to minimize the average network delay under any arbitrary spatial distribution of the ground users as well as the optimal cell partitions of UAVs and terrestrial base stations. The authors in [9] proposed

This research was supported by the ERC-PoC 727682 CacheMire project.the U.S. National Science Foundation under Grants CNS-1460316 and IIS-1633363.

a distributed cell association mechanism for energy harvesting IoT devices based on mean-field multi-armed games. In [11], the authors formulated the problem of cell association as a noncooperative game and proposed a distributed algorithm based on the machine learning framework of echo state networks (ESNs). The proposed algorithm enables the small base stations to autonomously choose their optimal bands allocation strategies while having only limited information on the states of the network and its users. The work in [12] also used machine learning to study cell association in cloud-based networks. Altough interesting, all these works either consider a static model or dynamic systems where all the information are assumed to be known to the BSs and users.

The main contribution of this paper is a novel collaborative learning mechanism in ultra-dense cellular networks that can exploit the similarities between users in terms of network conditions. To this end, we introduce a new learning mechanism *via imitation* that helps a user to select its serving BS faster by exploiting its local data and the learning outcomes of neighboring users. In fact, neighboring users might be characterized by similar characteristics such as channel conditions and their distance to the BSs. In this mechanism, instead of exchanging all the local data between users, only the outcome of their learning algorithms is transmitted.

In particular, we formulate the problem of cell association as a mean-field game (MFG) [15] with imitation in which a user aims to maximize its own data rate while minimizing the cost of imitating its neighboring users. Then, we reduce the MFG into a Markov decision problem (MDP) which is essential for exploiting the measurements available at the users. Hence, learning which base stations the users should connect to as well as the reward function via local data becomes possible.

To reach the desirable mean-field equilibrium outcome for the formulated game, we propose a deep-learning based reinforcement learning algorithm that allows the users to predict their utility function by exploiting their local data and mimicking similar users. Using extensive simulations, we compare the proposed mechanism with a setting in which users select their serving BSs without imitating other similar users. Such a comparison allows us to see that these imitator users can learn the optimal action in a new environment faster than other users.

The rest of this paper is organized as follows. In Section II, we present the system model. In Section III, the problem is formulated as a mean-field game with imitation and then a deep-learning based reinforcement learning algorithm is proposed to determine the user-cell association policy. Section IV presents the simulation results and Section V concludes the work.

II. System Model

Consider a set $\mathcal{S}$ of S small base stations (SBSs) deployed to serve a set $\mathcal{U}$ of U users in an LTE cellular system. We consider both downlink and uplink of the LTE system. We use $u \in \mathcal{U}$ and $s \in \mathcal{S}$ to index the users and SBSs, respectively. We introduce a binary variable a_{su} that is equal to 1 when user u chooses SBS s and 0 when user u is connected to another SBS. Each user u decides to select SBS s based on the following utility function:

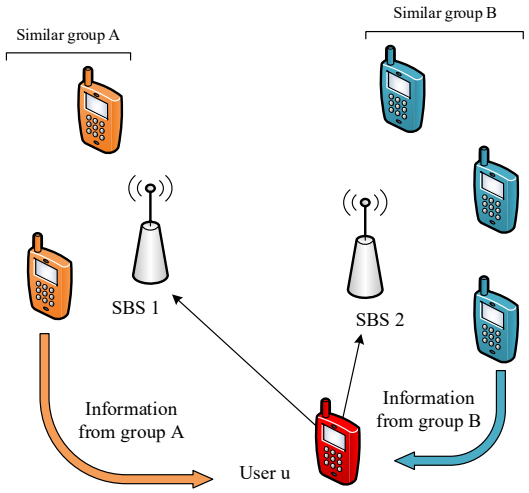

Fig. 1. Illustration of our system model.

$$a_{su} = \arg\max_{s} \left[r_{us} - \sum_{u'=1}^{U} f(u,u') |a_{su} - a_{su'}| \right], \quad (1)$$

where r_{us} is the throughput of user u when connected to SBS s, and f is a function that captures the similarity between two users u and u'. The similarity function captures the channel conditions, the geographical position and the interests of users.

The achievable throughput of user u is given by

$$r_{u,s} = B \log_2 \left(1 + \frac{p_{us} h_{us}}{\sigma^2 + I(\sum_{u'} a_{su'})} \right) - c(\sum_{u'=1}^{U} a_{su'}), \quad (2)$$

where $c(.)$ and $I(.)$ are increasing functions. $c(\sum_{u'=1}^{U} a_{su'})$ takes into account the throughput drop when the cell is congested. $I(\sum_{u'} a_{su'})$ determines interference in uplink and it is a function of the number of total users connected to the SBS u. h_{us} is the channel gain between user u and BS s. An illustration of the system model is shown in Fig. 1

In ultra-dense cellular systems, a large number of users deployed in the same area are covered by the same SBSs and experience the same quality-of-service (QoS) when the number of users tends to infinity. Thus, we introduce a similarity indicator function $f(u,u')$ and define it as follows:

$$f(u,u') = \begin{cases} 0, & \text{user } u \text{ is not similar to user } u', \\ 1, & \text{user } u \text{ is similar to user } u'. \end{cases} \quad (3)$$

The user distribution across SBSs at time slot t is given by:

$$\boldsymbol{\pi}(t) = \begin{bmatrix} \pi_1(t) & \cdots & \pi_S(t) \end{bmatrix}, \quad (4)$$

where $\pi_s(t)$ is the fraction of users that are connected to SBS j at time slot t. We can define transition probability $P_{sm}(t)$

as the probability that users connected to SBS s switch to SBS m in time slot t. Hence, the users' distribution evolves as follows:

$$\pi_m(t+1) = \sum_{s=1}^{S} P_{sm}(t)\pi_s(t). \quad (5)$$

The reward function for each user is defined as a function of $\boldsymbol{\pi}(t)$:

$$r_u(\pi_s(t), s) = r_{us}(\pi_s(t)) - \sum_{u'=1}^{U} f(u,u')|a_{su} - a_{Su'}|. \quad (6)$$

Our primary goal is to assign the users to the SBSs while accounting for the high density of users in cellular systems and leveraging this density. To this end, we formulate the assignment problem as a mean-field problem in which the users exploit the storage and computing capabilities at the users to cooperatively decide to which SBSs they connect.

III. Problem Formulation and Game Analysis

In this section, we formulate the problem of user-cell association as a mean-field game [13], [15] with imitation to account for the high density of future cellular systems and leverage the data available at the users with a low communication overhead. Thus, we enable the users to collaborate and leverage both storage and processing capabilities that are locally available to them for an efficient cell association mechanism.

A. Mean-Field Game Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph whose vertex set $\mathcal{V}$ represents the set of SBSs to which the users can connect and the edge set $\mathcal{E}$ represents the possible transition between every two SBSs. Thus, only the neighboring SBSs are connected with an undirected link. We define the state of a given user $u \in \mathcal{U}$ at time t as the SBS to which this user will get connected.

For each SBS $s \in \mathcal{S}$, we define the two sets $\mathcal{V}_s = \{j : (j,i) \in \mathcal{E}\}$ and $\bar{\mathcal{V}}_s = \mathcal{V}_s \cup \{s\}$. The dynamics of the users are generated by right stochastic matrices $\boldsymbol{P}(t) \in S(\mathcal{G})$, where $\mathbb{S}(\mathcal{G}) = \mathbb{S}_1(\mathcal{G}) \times ... \times \mathbb{S}_S(\mathcal{G})$ and each row $\boldsymbol{P}_s(t)$ belongs to $\mathbb{S}_s(\mathcal{G}) = \{p\Delta^{S-1} | \text{supp}(p) \subset \bar{\mathcal{V}}_s\}$, where Δ^{S-1} is the simplex in $\mathbb{R}^S$. Moreover, we define a value function $V_s(t)$ of state s at time t, and a reward function $r_s(\boldsymbol{\pi}(t), \boldsymbol{P}_s(t))$, quantifying the instantaneous reward of a user connected to SBS s taking transitions with probability $P_s(t)$ when the current distribution of the users over the SBSs is $\boldsymbol{\pi}(t)$.

The backward Hamilton Jacobi-Bellman (HJB) equation and the forward Fokker-Planck equation for each SBS $s \in \{1, ..., S\}$ and time $t = 0, ..., T-1$, in a discrete-time graph state MFG are given by:

$$V_s^t = \max_{\boldsymbol{P}_s^t \in S(\mathcal{G})} \left\{ r_s(\boldsymbol{\pi}(t), \boldsymbol{P}_s(t)) + \sum_{j \in \bar{\mathcal{V}}_s} \boldsymbol{P}_{sj}(t) V_j(t+1) \right\}, \quad (7)$$

$$\boldsymbol{\pi}_s(t+1) = \sum_{j \in \bar{\mathcal{V}}} \boldsymbol{P}_{js}(t) \boldsymbol{\pi}_s(t). \quad (8)$$

Next, we define the elements that are necessary to formulate our problem.

- *Users distribution* $\boldsymbol{p}_s(t) \in \Delta^{S-1}$ for $t = 0, ..., T-1$. Each $\boldsymbol{\pi}(t)$ is a discrete probability distribution over S SBSs, where $\boldsymbol{\pi}_s(t)$ is the fraction of users that are connected to SBS s at time t.
- *Transition matrix* $\boldsymbol{P}(t) \in S(\mathcal{G})$. $\boldsymbol{P}_{sj}(t)$ is the probability that users connected to SBS s switch to SBS j at time t. We refer to $P_s(t)$ as the action of users conneced to SBS s. $\boldsymbol{P}(t)$ generated the forward equation

$$\pi_j(t+1) = \sum_{s=1}^{S} \boldsymbol{P}_{sj}(t)\pi_s(t). \quad (9)$$

- *Reward* $r_s(\boldsymbol{\pi}(t), \boldsymbol{P}_s(t)) = \sum_{j=1}^{S} \boldsymbol{P}_{sj}(t) r_{sj}(\boldsymbol{\pi}(t), \boldsymbol{P}_s(t))$ for $s \in \mathcal{S}$. This is the reward received by the users connected to SBS s that choose action $\boldsymbol{P}_s(t)$ at time t, when the distribution is $\boldsymbol{\pi}(t)$.
- *Value function* $V(t) \in R^S$. $V_s(t)$ is the expected maximum total reward of being connected to SBS s at time t. A terminal value V^{T-1} is needed and will be set to zero.
- *Average reward* $e_s(\boldsymbol{\pi}, \boldsymbol{P}, V)$, for $s \in \mathcal{S}$ and $V \in R^S$ and $P \in S(\mathcal{G})$. This is the average reward received by users connected to SBS S when the current distribution is $\boldsymbol{\pi}$, action $\boldsymbol{P}$ is chosen, and the subsequent expected total reward is V. The average reward is defined as

$$e_s(\boldsymbol{\pi}, \boldsymbol{P}, V) = \sum_{j=1}^{S} \boldsymbol{P}_{sj}(r_{sj}(\boldsymbol{\pi}, \boldsymbol{P}) + V_j). \quad (10)$$

Intuitvely, users want to act optimally in order to maximize their expected total average reward.

For $P \in S(\mathcal{G})$ and a vector $q \in S_s(\mathcal{G})$, we let $\mathcal{P}(\boldsymbol{P}, s, q)$ be the matrix equal to P, where row s is replaced by q. Then, we define the desirable outcome of the problem as follows.

Definition 1: A right stochastic matrix $P \in \mathbb{S}(\mathcal{G})$, is a Nash maximizer of $e(\boldsymbol{\pi}, P, V)$, if given a fixed $\boldsymbol{\pi}$ and a fixed $V \in \mathbb{R}^S$, for any $s \in \mathcal{S}$ and any $q \in \mathbb{S}_s(\mathcal{G})$, there is

$$e_s(\boldsymbol{\pi}, \boldsymbol{P}, V) \geq e_s(\boldsymbol{\pi}, \mathcal{P}(\boldsymbol{P}, s, q), V). \quad (11)$$

The rows of $\boldsymbol{P}$ form a Nash equilibrium set of actions, since for any SBS s, the users connected to SBS s cannot increase their reward by unilaterally switching their action from $\boldsymbol{P}_s$ to any q. Under Definition 1, the value function of each SBS s at each time t satisfies the optimality criteria:

$$V_s(t) = \max_{q \in S_s(\mathcal{G})} \left(\sum_{j=1}^{S} q_j \left[r_{sj}(\boldsymbol{\pi}(t), \mathcal{P}(\boldsymbol{P}(t), s, q)) + V_j(t+1) \right] \right). \quad (12)$$

A solution of the MFG is a sequence of pairs $\{(\boldsymbol{\pi}(t), V(t))\}_{t=0,...,T}$ satisfying the optimality criteria (12) and the forward equation (9).

Now, we reduce the formulated MFG into a single user deterministic Markov decision process (MDP) within a finite time duration. This shows that solving the optimization problem of a single user MDP is equivalent to solving the MFG and allowing every user to select the SBS that maximizes its value function. This connection will allow us to apply efficient deep reinforcement learning (RL) methods that use data about the dynamics of the users in the network, to learn the best strategies of the users as well as their reward function.

B. Mean-Field Game Analysis

Here we formulate the problem as a Markov decision process for each user. Each user's action is defined as choosing an SBS to connect to. Its reward is defined in (6). Also the state of the system $\boldsymbol{x}^t$ is defined as the number of users connected to each SBS:

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) & \cdots & x_S(t) \end{bmatrix}, \tag{13}$$

First, we need to find number of states for the MDP as follows.

Proposition 1: Let the number of SBSs that each user u can use be $b_u \leq S$. Also, let the total number of users that can connect to SBS s be N_s. The total number of states for user u is S_u and is bounded as follows:

$$K_u \leq \binom{\sum_{s=1}^{b_u} N_s - b_u + 1}{b_u - 1} \leq \binom{U - b_u + 1}{b_u - 1} \tag{14}$$

Proof: The total number of states for a given user u is given by the non-negative integral solutions of the following equation:

$$K_u = \sum_{s=1}^{b_u} n_s = \sum_{s=1}^{b_u} N_s, \tag{15}$$

where n_s is actual number of users connected to SBS s. Hence, we can find the upper limit for K_u as,

$$K_u \leq \binom{\sum_{s=1}^{b_u} N_s - b_u + 1}{b_u - 1}. \tag{16}$$

Furthermore, since

$$\sum_{s=1}^{b_u} N_s \leq U, \tag{17}$$

we know that

$$\binom{\sum_{s=1}^{b_u} N_s - b_u + 1}{b_u - 1} \leq \binom{U - b_u + 1}{b_u - 1}. \tag{18}$$

■

As we can see from Proposition 1, the size of the state space grows with the number of users in the order of $\mathcal{O}(U^{b_u})$ in the worst case. Since each agent uses Q-learning for learning the optimal action, it has to store the Q-function. However, as we can see from Proposition 1, it is not feasible to create a table for the Q-function. The only assumption we make on the system is that each user knows its reward after connecting to each SBS. A user can estimate $\sum_{u'=1}^{U} a_{su'}$ based only on its own reward. We know

$$\frac{\partial r_{u,s}}{\partial \sum_{u'=1}^{K} a_{su'}} = -c'\Big(\sum_{u'=1}^{U} a_{su'}\Big) - \frac{p_{ij}h_{ij}I'\big(\sum_{u'=1}^{K} a_{su'}\big)}{\big(\sigma^2 + I\big(\sum_{u'=1}^{K} a_{su'}\big)\big)\big(\sigma^2 + I\big(\sum_{u'=1}^{U} a_{su'}\big) + p_{ij}h_{ij}\big)}, \tag{19}$$

and we know that $\frac{dI(x)}{x} > 0$ and $\frac{dc(x)}{x} > 0$. Hence, using the bisection method with the knowledge of $r_{u,s}$ each user can find total number of users connected to the BS.

Since each user cannot observe the full state of the system and it only observes it partially, we propose a method to solve partially observable Markov decision processes (POMDPs). In this method, each user estimates the full state using its limited observations and a neural Q-network. We use multilayer neural networks as Q-function estimator. At each time slot t, all the users make decision using reinforcement learning and estimate their Q-function using multilayer neural network.

C. Value function approximation

As we showed, the value function cannot be stored in a table. Therefore, a function approximation method should be used to approximate the value function. Neural networks are powerful tools for value function approximation [16].

Since each user does not know the transition model of the MDP, they need to approximate the Q-function instead of the value function. Considering the fact that training adaptive linear neurons using the backpropagation algorithm is computationally inexpensive, we can use a unique neural network n for approximating $Q(e, a_n)$. e is the partial state observed by the user in the system and a_n is the action, i.e., the SBS selected by the user. In this neural network, the state of the system is the input to the neural network. This is due to the fact that the number of actions for each user is limited in contrast to the large number of states.

The output of each adaptive linear neuron can be written as

$$Q_u(e, a_n) = \boldsymbol{w}_n^T \boldsymbol{x} + \boldsymbol{b}_n. \tag{20}$$

The approximating process is to choose a random action at each stage and then trying to update the weights. To do so, we update the weights based on the following rule:

$$\boldsymbol{w}_n(t+1) = \boldsymbol{w}_n(t) + \lambda(y_u - Q_u(e, a_n)), \tag{21}$$

where λ is the learning rate, and y is the current target which is an exponential moving average and can be written as

$$y_u = (1-\alpha)Q_u(e, a_n) + \alpha(r(e, a_n) + \gamma \max_n \boldsymbol{w}_n^T \boldsymbol{x} + \boldsymbol{b}_n), \tag{22}$$

where α is a factor between 0 and 1. We use these adaptive linear neurons in a multilayer structure to estimate the Q-function, using which we find the optimal action as follows:

$$a_u(e) = \max_n Q_u(e, a_n). \tag{23}$$

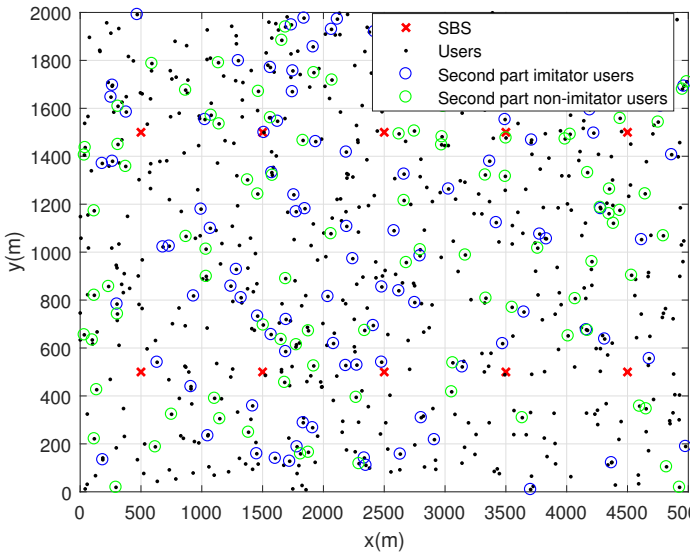

Fig. 2. Wireless network with 10 SBS and 700 users.

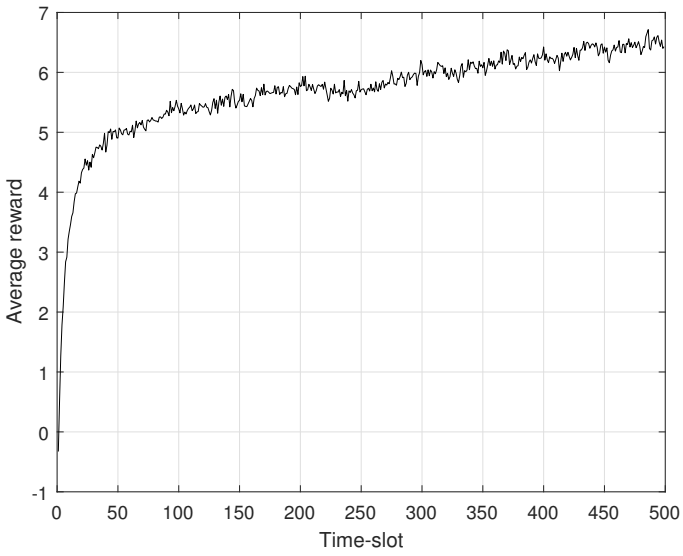

Fig. 3. Average reward for 500 users in the first phase.

That is, each user u finds its best action $a_u(e)$ in state e using the maximum output of its neural networks. Each neural network estimates the value of an action in the current partially observed state.

IV. Simulation Results and Analysis

For our simulation, we consider a network with 700 users uniformly distributed in the range of 10 SBSs.

We assume that the path loss exponent is 2, the carrier frequency is 900 MHz, and the noise variance is -173.9 dbm/Hz. Each user acts based on an ϵ-greedy reinforcement learning, meaning that it chooses a random action with probability ϵ and approximates the value function using adaptive linear neuron (ADALINE) neural networks. The user, then, trains its network using the Widrow-Hoff algorithm (exploration) and chooses the best BS with probability $1-\epsilon$ using [23] (exploitation).

There are two different types of users in the learning algorithm:

1) *Non-imitator users*: users that maximize a reward function affected only by the throughput and congestion functions.
2) *Imitator users*: users that maximize the reward function of non-imitator users in addition to imitating the action of the users close to them.

The simulation consists of two phases. In the first phase, 500 non-imitator users start to approximate and maximize their value function using the aforementioned methods. After they converge to an equilibrium, in the second phase, 100 non-imitator and 100 imitator users enter the system. As we can see in Fig. 3, non-imitator users start to learn the environment in 500 time-slots and their average reward increases with time. This is due to the fact that, as they learn to coordinate with each other and manage interference, their average reward will increase. Fig. 4 shows the second phase of the simulation where 100 imitator users and 100 non-imitator users enter the system and start to approximate a value function and maximize it. The locations of the users, imitator users, non-imitator users and SBSs in the system are depicted in Fig. 2.

As we can see, the imitator users adapt faster to the environment. This is due to the fact that, in addition to learning the environment, the imitator users also use the existing users' experience. This experience will help them to learn the environment and behavior of other users faster. Since non-imitator users will also gain experience over time, the average reward of non-imitator users will eventually approach that of imitator users. However, non-imitator users gain the experience of the existing users with a delay which is in a direct relationship with the experience of their adjacent users. The average number of users per base station in two different cases is shown in Fig. 5. After 500 iterations, in case 1, 200 imitator users enter the system, and in case 2, 200 non-imitator users enter the system. Then, we find the average number of users per base station for 100 time-slots. Since the imitator users use the existing knowledge of 500 users in the system, they can learn to adapt with the system faster, and as a result, the load is more evenly balanced in this case.

V. Conclusion

In this paper, we have addressed the problem of cell association in ultra-dense networks while leveraging the data available at the users and their processing power. First, we have formulated the problem as a mean-field game with imitation, where users not only learn from their own local data, but also from the models learned by their neighboring users with the same characteristics, captured by a similarity function. We have showed via simulations that the proposed collaborative learning mechanism outperforms the learning

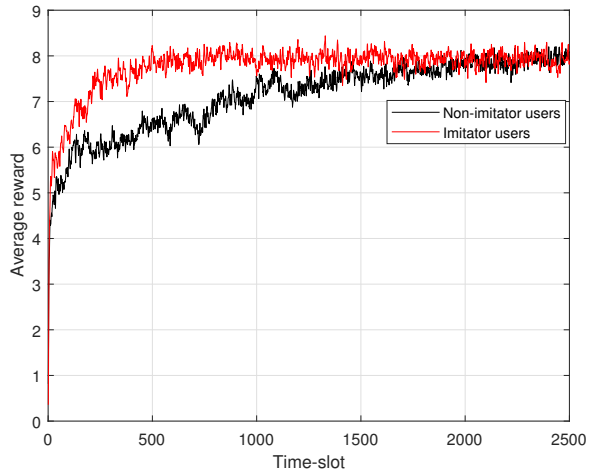

Fig. 4. Transient time of average reward for 100 imitator and 100 non-imitator users in the second phase.

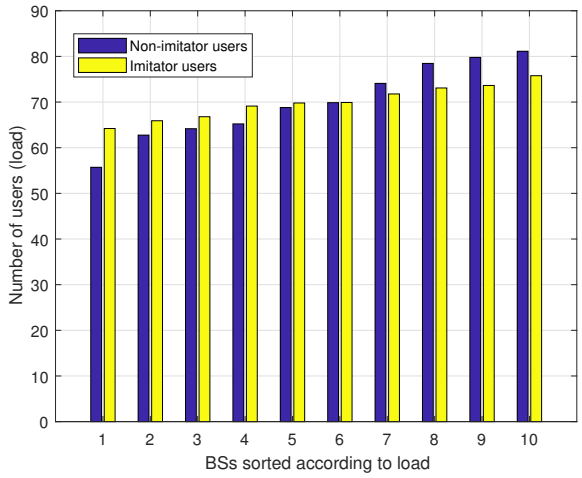

Fig. 5. Load balance for each SBS in case of imitator and non-imitator users.

mechanism without imitation in terms of learning time and load balance.

REFERENCES

[1] T. Park, N. Abuzainab, and W. Saad, "Learning How to Communicate in the Internet of Things: Finite Resources and Heterogeneity", *IEEE Access, Special Issue on Optimization for Emerging Wireless Networks: IoT, 5G and Smart Grid Communication Networks*, vol. 4, November 2016.

[2] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani, Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges, *IEEE Wireless Communications*, vol. 24, no. 3, pp. 1016, June 2017.

[3] N. C. Luong, D. T. Hoang, P. Wang, D. Niyato, D. I. Kim, and Z. Han, Data collection and wireless communication in internet of things (IoT) using economic analysis and pricing models: a survey, *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 25462590, June 2016.

[4] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned Aerial Vehicle with Underlaid Device-to-Device Communications: Performance and Tradeoffs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949 - 3963, June 2016.

[5] N. Abuzainab, W. Saad, C. S. Hong, and H. V. Poor, "Cognitive Hierarchy Theory for Distributed Resource Allocation in the Internet of Things", *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 7687 - 7702, December 2017.

[6] T. OShea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no 4, p. 563-575, 2017.

[7] H. Elshaer, M. N. Kulkarni, F. Boccardi, J. G. Andrews, and M. Dohler, Downlink and uplink cell association with traditional macrocells and millimeter wave small cells, *IEEE Transactions on Wireless Communications*, Vol. 15, No. 9, pp. 6244 - 6258, 2016.

[8] M. Mozaffari, W. Saad, M. Bennis and and M. Debbah, "Optimal Transport Theory for Cell Association in UAV-Enabled Cellular Networks", *IEEE Communications Letters*, Vol. 21, No. 9, pp. 2053-2056, 2017.

[9] S. Maghsudi and E. Hossain, "Distributed Cell Association for Energy Harvesting IoT Devices in Dense Small Cell Networks: A Mean-Field Multi-Armed Bandit Approach", *IEEE Access*, Vol. 5, 3513-3523, 2017.

[10] M. Chen, U. Challita, W. Saad and M. Debbah, "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks", *arXiv preprint arXiv:1710.02913,* 2017.

[11] M. Chen, W. Saad, and C. Yin, "Echo State Networks for Self-Organizing Resource Allocation in LTE-U with Uplink-Downlink Decoupling", *IEEE Transactions on Wireless Communications,* vol. 16, no 1, p. 3-16, 2017.

[12] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the Sky: Proactive Deployment of Cache-Enabled Unmanned Aerial Vehicles for Optimized Quality-of-Experience", *IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Human-In-The-Loop Mobile Networks*, vol. 35, no. 5, pp. 1046 - 1061, May 2017.

[13] H. B. McMahan, E. Moore, D. Ramage, S. Hampson B. A. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data", *arXiv preprint arXiv:1602.05629*, 2016.

[14] J. Konecny, H. B. McMahan, D. Ramage, P. Richtrik, "Federated Optimization: Distributed Machine Learning for On-Device Intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[15] O. Guéant, J-M. Lasry, P-L. Lions, Mean field games and applications, *Paris-Princeton lectures on mathematical finance 2010*, 2011, pp. 205-266.

[16] 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, 02 2015.

[17] A. T. Z. Kasgari, W. Saad, and M. Debbah, "Brain-aware wireless networks: Learning and resource management," in *Proc. 51th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA*, Nov 2017.