Reinforcement Learning for User Association and Handover in mmWave-Enabled Networks

Alireza Alizadeh, *Member, IEEE*, and Mai Vu, *Senior Member, IEEE*

***Abstract*— Using a multi-armed bandit technique, we propose centralized and semi-distributed online algorithms for load balancing user association and handover in mmWave-enabled networks. Load balancing at all base stations (BSs) imposes explicit constraints that makes the actions of all user equipment (UEs) co-dependent, a challenging twist to reinforcement learning. We propose a central load balancer to guarantee load balancing at all BSs for every learning step. We consider two association vectors: one for leaning update, and one best-to-date for data transmission, allowing UEs to engage in best-result data transmission while effectively participating in a background learning process indefinitely. For dynamic networks, we introduce a measurement model capturing rapid channel variations and user mobility. To minimize handover rate, we also differentiate between the handover cost for transmission and that for learning, and introduce a learning handover cost decreasing with sojourn time. The proposed algorithms can be implemented online as they require no offline training and can effectively adapt to network dynamics. Numerical results show that the proposed algorithms exhibit fast learning convergence and outperform 3GPP handover by achieving an order of magnitude lower handover rate at a significantly higher network sum-rate, reaching within 94-97% of the near-optimal worst connection swapping benchmark algorithm.**

***Index Terms*— Multi-armed bandit, user association, load balancing, handover, mmWave-enabled cellular networks.**

I. Introduction

THE ever-increasing demand for higher data rates requires deployment of dense cellular networks with coexistence of sub-6 GHz macro base stations (MBSs) and millimeter wave (mmWave) small base stations (SBSs). A challenging problem in these dense networks is load balancing user association: finding the best connections between BSs and user equipment (UEs) to achieve an optimal network performance while balancing the BSs' loads (maximum number of data streams transmitted by each BS). In a mobile mmWave-enabled network, this problem integrates with handover and becomes even more complicated.

Coordinated multipoint (CoMP) allows a UE to connect with multiple cooperating BSs, simultaneously. For downlink CoMP, there are two main categories: 1) joint processing, and 2) coordinated scheduling/beamforming (CS/CB). In the joint processing, data for each UE is available at more than one BS, while in CS/CB, the data for each UE is only available at and transmitted from one BS; however, the cooperating BSs jointly make decisions about user scheduling/beamforming. In this paper, we focus on the first approach which improves the data rate of cell-edge UEs, while at the same time, complicate the user association and handover problems [2].

Manuscript received 28 November 2020; revised 11 June 2021, 1 December 2021, and 26 March 2022; accepted 13 May 2022. Date of publication 7 June 2022; date of current version 11 November 2022. This work was supported in part by the National Science Foundation under CNS Grant 1908552. The associate editor coordinating the review of this article and approving it for publication was L.-C. Wang. *(Corresponding author: Alireza Alizadeh.)*

The authors are with the Department of Electrical and Computer Engineering, Tufts University, Medford, MA 02155 USA (e-mail: alireza.alizadeh@tufts.edu; mai.vu@tufts.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2022.3178767.

Digital Object Identifier 10.1109/TWC.2022.3178767

A. Related and Prior Work

The problem of unique user association has attracted the attention of many researchers due to its importance and impact on network performance. In this problem, each UE can only connect to a BS at each time instant. Load balancing unique user association results in a complex integer non-linear programming which is usually NP-hard [3]. Theoretical approaches and algorithms have been designed to solve this problem and achieve near optimal solutions [3]–[5]. An analytical solution for this problem is proposed in [3], where the authors relaxed unique association constraints to solve the problem and then used a rounding method to convert to integer association variables. A similar approach for massive MIMO networks is studied in [4]. For a 60 GHz wireless network, the user association is studied in [5], where the authors assumed negligible interference due to highly directional transmissions. This assumption, however, becomes inaccurate for mmWave-enabled cellular networks as they can transit from noise-limited to interference-limited regimes [6]. In [7], the authors proposed a heuristic algorithm for load balancing user association, called worst connection swapping (WCS), which achieves near-optimal performance by taking into account the dependency of user association and interference. The WCS algorithm has been used as a near-optimal benchmark performance in [8], [9].

The aforementioned works studied the user association problem in an static setting without user mobility. In a highly dynamic mmWave-enabled network, however, an efficient user association algorithm must take into account both the effects of abrupt channel variations and user mobility. The 3GPP user association and handover mechanism is based on maximum signal to interference and noise ratio (max-SINR) strategy, where each UE associate with the best available BS providing the highest SINR value. This approach, however, is not efficient for B5G mmWave-enabled networks as it incurs large number of handovers and thus significant signaling overhead due to abrupt mmWave channel variations [10]. In mmWave-enabled networks, the UE may remain associated with a BS for a very short period of time, as little as 0.75 s [11].

In recent years, reinforcement learning (RL) emerges as a potential technique for effective user association in

cellular networks. RL is a powerful machine learning technique which can learn from real-time interaction with the system without requiring extensive training, making it particularly suitable for dynamic wireless systems. In RL, a learning agent interacts with its environment to receive a reward, and take an action to achieve an optimal policy. Distributed learning algorithms, including federated learning and multi agent-RL in particular, have a strong potential to be applied in wireless networks to solve many existing challenges [12]. For example, a distributed RL algorithm can be used to find optimal trajectories maximizing the coverage of ground users served by a set of drone BSs [13]. RL provides a solution for this non-convex optimization problem in a dynamic and unpredictable environment.

With regards to user association and handover problems, deep RL has been used for user association in dense and static networks [14], [15], and multi-agent Q-learning for joint power optimization and user association [16]. Deep learning is also used to perform online user association in massive MIMO 4G networks, where inputs to the neural network are only UEs locations [17]. For mobile mmWave networks, user association is formulated as a non-convex optimization and a deep deterministic gradient method is employed to approximate its in [9]. In these works, load balancing is commonly imposed only on SBSs but not on the MBS, allowing fully loaded SBS to offload to the MBS [14]–[16]. Alternatively, load constraints are formulated in a user association optimization, and a relaxed version is solved using standard linear programming to build a training set, which is then used to train a neural network with quota of BSs as inputs and optimal association as outputs [9], [17]. All these works, however, did not enforce the load balancing at all base stations and hence left open the question whether reinforcement learning could be used effectively for user association with load balancing.

In order to show that reinforcement learning can be effective for user association and handover under load balancing, we focus on multi-agent multi-armed bandit (MA-MAB), a simple but powerful RL framework in which, given a set of actions, each agent takes an action at each learning step and receives a reward in progress towards a goal. The agents have no prior knowledge about the rewards; however, at each learning step, each agent explores and exploits different actions, receives instantaneous rewards, and updates their expected average rewards in order to find their best possible next actions [18]. In general RL, there is a notion of the *state* perceived by each agent about the environment, and transition of each agent to the next state depends on the current state and the action taken; however, MAB is a classical stateless RL in which the environment is not associated with any state. An MA-MAB framework has been used for dynamic spectrum scheduling [19], where the agents used the upper confidence bound (UCB) action selection rule to either explore a new channel to learn more about channel statistics, or exploit the known best channel (with the highest expected reward) to minimize the expected total regret. An MAB technique is also utilized to solve a joint beam tracking and adaptive rate selection problem in mmWave networks [20], improving the throughput by up to 182% compared to static beam management proposed for 5G NR [21].

In the context of user association, a UE can be considered as an agent and selecting a BS can be considered as taking an action. Thus, a user association problem can be cast as an MA-MAB. MAB techniques have been applied for user association in LTE networks, but without load balancing and user mobility, where each UE tend to connect to the BS which provides the highest average reward regardless of BSs' loads [22]. MAB is employed to mitigate frequent handover in dense cellular networks [23], [24], where only single-agent MAB technique for a typical UE is applied to study the handover problem. MA-MAB techniques have not been applied to wireless handover problem. Furthermore, with load balancing constraints at all BSs, the actions of all agents (UEs) are interdependent, which introduces new complexity and has not been considered using RL.

B. Our Contributions

We propose MAB-based centralized and semi-distributed algorithms for load balancing user association and handover in mmWave-enabled networks with CoMP technology, applicable to B5G/6G systems. Load-balancing conditions impose coupling and inter-dependency between all users' actions and are enforced by a central entity, which uses reward information of all users to assign BS connections to satisfy the load constraints and achieve a high total reward. The proposed algorithms can be implemented online as they require no training and can efficiently adapt to network dynamics by exhibiting fast convergence. These algorithms allow UEs to engage in data transmission while effectively participating in a learning process which continues indefinitely in the background. To the best of our knowledge, this is the first work that employs MAB techniques for online user association and handover with explicit load balancing at all BSs. The main contributions of this paper are:

i) Our learning algorithm ensures load balancing at all BSs in all tiers (instead of just the small BSs as considered in the literature) by considering a specific quota for each MBS and each SBS separately. Load balancing at all BSs introduces inter-dependency among the actions of all UEs. In existing learning-based approaches, each UE tends to connect with the BS providing the highest reward independently of the actions of other UEs, which can cause a collision if the number of UEs simultaneously selecting a BS is more that its quota. To overcome collision and the associated penalty in such a multi-agent RL setting, we propose a *joint-UCB (J-UCB) action selection rule* and introduce a central entity, the *central load balancer (CLB)*, to assign association to obtain a high network sum-rate and guarantee load balancing. This CLB handles all the dependency among UEs' actions and removes any need for UEs to exchange messages among themselves. To the best of our knowledge, our work thus provides the first learning solution to this difficult problem of guaranteeing BSs load balancing in user association.

ii) Employing the CLB, we propose both a centralized and a semi-distributed MAB algorithms, using only local measurements to compute the reward at each UE and do not require full channel state information. Actions are chosen by the CLB in the centralized approach, and by the UEs in

the semi-distributed one. In both approaches, we differentiate between the learning and the transmission processes, and use two association vectors: one for learning update, and one as the best-to-date association for transmission. These two association vectors allow the learning process to utilize its history while using the best learned results for transmission.

iii) We propose a number of innovations for handover, including a measurement model which captures the underlined user mobility, and the concept of handover cost for learning that is different from handover cost for data transmission. We also propose a learning handover cost model that has decreasing cost the longer the UE has connected to the current BS. We introduce two updating rules with regards to the learning and the best-to-date association vectors, and apply different updating frequencies to utilize the latest measurements. Real-time learning carries out a single update with each new measurement, whereas background learning allows the updating to continue in the background in between measurements in order to improve the reward estimates. These different learning algorithms produce interesting trade-offs between the handover rate and the network throughput.

iv) The proposed algorithms allow online implementation since the learning process can continue indefinitely in the background, while UEs are engaged in data transmission according to the best-to-date association vector. Our results show fast convergence of both the proposed centralized and semi-distributed algorithms in a static network. For a dynamic networks, our results also confirm their efficacy and show that the proposed online centralized MAB algorithm outperforms 3GPP handover algorithm in terms of both network throughput and handover rate. Compared to a near-optimal non-learning centralized solution which requires full CSI of all UEs and BSs at the central entity [7], our learning algorithms reach closely within 94-97% its performance with only local measurements at each UE.

C. Notation

We denote the scalars and sets by italic letters (e.g. x or X) and calligraphy letters (e.g. $\mathcal{X}$), respectively. $|\mathcal{X}|$ denotes the cardinality of set $\mathcal{X}$. Vectors are represented by lowercase boldface letters (e.g. $\mathbf{x}$), and matrices by uppercase boldface letters (e.g. $\mathbf{X}$). Superscript $(.)^T$ and $(.)^*$ represent the transpose operator and the conjugate transpose operator, respectively. $\log(.)$ stands for base-2 logarithm, and big-O notation $\mathcal{O}(.)$ expresses the complexity. We define the delta functions $\delta(x,y)$ and $\bar{\delta}(x,y)$ such that $\delta(x,y)=1$ if $x=y$ and $\delta(x,y)=0$ if $x\neq y$, and $\bar{\delta}(x,y)=1-\delta(x,y)$.

II. System Model

We study the problem of user association in a multi-tier HetNet with MBSs operating at a microwave (sub-6 GHz) band and SBSs working at a mmWave band. In this section, we introduce the network, channel, and signal models.

A. Network and Channel Models

We consider the downlink of a two-tier cellular HetNet with J_B MBSs in tier-1, J_S SBSs in tier-2, and K UEs. Let $\mathcal{J}_B$, $\mathcal{J}_S$, and $\mathcal{J}=\{1,\ldots,j,\ldots,J\}$ denote the respective sets of MBSs, SBSs, and all BSs with $J=J_B+J_S$, and $\mathcal{K}=\{1,\ldots,k,\ldots,K\}$ represents the set of UEs. Each UE k can request up to n_k data streams for data transmission and receive data from multiple BSs via joint transmission CoMP technique [2]. This is different from fractional association as defined in [4], where association variables are not integer and represent long-term average associations. Instead, we allow each UE k to receive multiple data streams from multiple BSs, and assume it is composed of n_k virtual UEs (VUEs) each requesting a single data stream. Moreover, we define $\mathcal{V}=\{1_1,\ldots,1_{n_1},\ldots,k_1,\ldots,k_{n_k},\ldots,K_1,\ldots,K_{n_K}\}$ as the set of VUEs with $|\mathcal{V}|=\sum_{k=1}^{K} n_k$, and allow each VUE k_v ($v\in\{1,2,\ldots,n_k\}$) to associate with either a MBS or a SBS.

Each BS j is equipped with a uniform planar array (UPA) antenna with M_j elements, and has a specific *quota* q_j which represents the maximum number of downlink data streams it can transmit. We assume $1\leq q_j\leq M_j$, where the upper bound is due to the fact that the number of data streams transmitted by each BS cannot exceed the number of its antennas. We also define the *quota vector* of BSs as $\mathbf{q}=[q_1,\ldots,q_J]$. Each UE k is equipped with an antenna array with $N_k^{\mu\mathrm{W}}$ elements for sub-6 GHz band and an antenna array with N_k^{mmW} elements for mmWave band such that $1\leq n_k\leq N_k^{\mu\mathrm{W}}+N_k^{\mathrm{mmW}}$.

In the sub-6 GHz band, we use the well-known Gaussian MIMO channel model. Denote $\mathbf{H}_{k,j}^{\mu\mathrm{W}}\in\mathbb{C}^{N_k^{\mu\mathrm{W}}\times M_j}$ as the channel matrix between MBS j and UE k where the entries are i.i.d. complex Gaussian random variables given by $H^{\mu\mathrm{W}}\sim\mathcal{CN}(0,1)$. In the mmWave band, the transmissions are highly directional and the simple Gaussian MIMO channel may not hold. Instead, we employ the $N_k^{\mathrm{mmW}}\times M_j$ clustered mmWave MIMO channel between SBS j and UE k which includes C clusters with G rays per cluster defined as [25], [26]

$$\mathbf{H}_{k,j}^{\mathrm{mmW}}=\frac{1}{\sqrt{CG}}\sum_{c=1}^{C}\sum_{g=1}^{G}\sqrt{\gamma_c}\;\mathbf{a}(\phi_k^{c,g},\theta_k^{c,g})\;\mathbf{a}^*(\phi_j^{c,g},\theta_j^{c,g}) \quad (1)$$

where γ_c is the power gain of the cth cluster. The parameters ϕ and θ represent azimuth and elevation angles, respectively. The vector $\mathbf{a}(\phi,\theta)$ is the response vector of a uniform planar array (UPA) which allows 3D beamforming in both the azimuth and elevation directions. We consider the probability of LoS and NLoS as given in [27], and utilize the path loss model for LoS and NLoS links as given in [25].

B. Signal Model

We assume all BSs and UEs, regardless of their operating band, are capable of performing beamforming technique which is introduced for both LTE [28] and 5G new radio (NR) [29]. Thus, BSs use precoders for data transmission and UEs employ combiners for data reception.

The effective interfering channel gain on VUE k_v from BS j while serving VUE l_u can be expressed as

$$h_{k_v,l_u,j}=\mathbf{w}_{k_v}^{*}\mathbf{H}_{k,j}\mathbf{f}_{l_u,j} \quad (2)$$

where $\mathbf{f}_{l_u,j}\in\mathbb{C}^{M_j\times 1}$ is the linear precoder (transmit beamforming vector) at BS j intended for VUE l_u, and $\mathbf{w}_{k_v}\in\mathbb{C}^{N_k\times 1}$ is the linear combiner (receive beamforming vector)

of VUE k_v. Thus, the effective channel gain between BS j and VUE k_v is $h_{k_v,j} = \mathbf{w}_{k_v}^* \mathbf{H}_{k,j} \mathbf{f}_{k_v,j}$.

Next, we define $\mathcal{K}_j$ as the *activation set of BS* j which represents the set of all active VUEs in BS j. Thus, the $M_j \times 1$ transmitted signal from BS j is given by

$$\mathbf{x}_j = \sum_{k_v \in \mathcal{K}_j} \mathbf{f}_{k_v,j} s_{k_v,j} \tag{3}$$

where $s_{k_v,j} \in \mathbb{C}$ is the transmitted symbol from BS j to VUE k_v, such that $\mathbb{E}[s_{k_v,j} s_{k_v,j}^*] = P_{k_v,j}$, and $P_{k_v,j}$ is the transmit power from BS j dedicated to VUE k_v.

Thus, the post-processed received signal (after combiner) at VUE k_v connected to BS j can be written as

$$y_{k_v} = \sum_{j \in \mathcal{J}} \sum_{l_u \in \mathcal{K}_j} h_{k_v,l_u,j} s_{l_u,j} + \mathbf{w}_{k_v}^* z_{k_v} \tag{4}$$

where $z_{k_v} \sim \mathcal{CN}(0, N_0)$ is the complex additive white Gaussian noise at VUE k_v, and N_0 is the noise power.

The presented signal model is applicable for all types of transmit beamforming and receive combining. In mmWave MIMO systems, hybrid beamforming can be implemented to reduces the number of RF chains and thus control the relevant cost and power consumption. In this paper, we employ SVD beamforming technique to obtain the precoder and combiner vectors at BSs and UEs, respectively [7].

C. User Association and Transmission Rate

We consider a learning-based user association approach. The proposed approach is online in the sense that while UEs are engaged in data transmission, the learning process continues indefinitely in the background, to update the best-to-date association vector for data transmission. Thus, we introduce two association vectors to distinguish between the learning process and data transmission process: 1) *learning association vector* $\boldsymbol{\eta}^{(t)}$, and 2) *best-to-date association vector for data transmission* $\boldsymbol{\beta}$. The superscript t in $\boldsymbol{\eta}^{(t)}$ represents the learning time step. During the learning process, the UE may connect to BSs in $\boldsymbol{\eta}^{(t)}$ briefly (in a much shorter time scale compared to transmission time) to collect rewards for learning purposes. After that, the UEs switch their associations to the best-to-date $\boldsymbol{\beta}$ for transmission. Thus, associations for learning purposes can alter as fast as every time step t, while associations for data transmission change whenever there is a better association vector. This learning model makes our user association scheme suitable for highly dynamic mmWave-enabled cellular networks, while the transmissions remain at the best-to-date rates.

The learning associations between VUEs and BSs at learning time step t can be defined by association vector

$$\boldsymbol{\eta}^{(t)} \triangleq [\boldsymbol{\eta}_1^{(t)}, \ldots, \boldsymbol{\eta}_K^{(t)}]^T, \tag{5}$$

where $\boldsymbol{\eta}_k^{(t)}$ represents the learning association vector of UE k defined as

$$\boldsymbol{\eta}_k^{(t)} \triangleq [\eta_{k_1}^{(t)}, \ldots, \eta_{k_{n_k}}^{(t)}]^T \tag{6}$$

and $\eta_{k_v}^{(t)}$ denotes the index of BS to which user k is associated with for receiving data stream v during learning time step t.

We can define the *load balancing* constraint for BS j as

$$\sum_{k_v \in \mathcal{V}} \mathbb{1}_{k_v,j}^{(t)} \leq q_j \tag{7}$$

where $\mathbb{1}_{k_v,j}^{(t)} = 1$ if $\eta_{k_v}^{(t)} = j$, and $\mathbb{1}_{k_v,j}^{(t)} = 0$ if $\eta_{k_v}^{(t)} \neq j$. This constraint indicates that the total number of data streams requested by VUEs associated with BS j at learning time step t, cannot exceed the maximum number of downlink data streams of BS j.

Due to highly directional links between BSs and UEs, fast-varying nature of mmWave channels, and their short coherence time, there is a dependency between user association and interference structure in mmWave cellular systems [7]. Considering the received signal y_{k_v} in (4), the *learning instantaneous rate* (instantaneous reward) of VUE k_v from BS j at learning time step t is obtained as

$$R_{k_v,j}(\boldsymbol{\eta}^{(t)}) = B_j \log_2 \left(1 + \frac{P_{k_v,j} h_{k_v,j} h_{k_v,j}^*}{I_{k_v,j}}\right) \tag{8}$$

where B_j is the bandwidth of BS j, and $I_{k_v,j}$ is the interference plus noise given as

$$\begin{aligned} I_{k_v,j} = & \sum_{\substack{k_u \in \mathcal{K}_j^{(t)} \\ u \neq v}} P_{k_u,j} h_{k_v,k_u,j} h_{k_v,k_u,j}^* \\ & + \sum_{\substack{l_u \in \mathcal{K}_j^{(t)} \\ l \neq k}} P_{l_u,j} h_{k_v,l_u,j} h_{k_v,l_u,j}^* \\ & + \sum_{\substack{i \in \mathcal{J} \\ i \neq j}} \sum_{l_u \in \mathcal{K}_i^{(t)}} P_{l_u,i} h_{k_v,l_u,i} h_{k_v,l_u,i}^* + N_0 B_j \mathbf{w}_{k_v}^* \mathbf{w}_{k_v} \end{aligned} \tag{9}$$

where the first, second and third terms represent inter-stream, intra-cell and inter-cell interferences, respectively. The presence of the activation set of BSs ($\mathcal{K}_j^{(t)}$) in $I_{k_v,j}$ indicates the dependency of interference and user association. We note that, for example, if VUE k_v is associated with BS $j \in \mathcal{J}_B$, the interference at the VUE comes from the BSs in the same tier, i.e., $i \in \mathcal{J}_B$.

The overall learning instantaneous transmission rate of physical UE k is obtained by summing over all the VUEs corresponding to that physical UE. The total instantaneous reward computed as the overall network *sum-rate* given by

$$r(\boldsymbol{\eta}^{(t)}) = \sum_{k=1}^{K} \sum_{v=1}^{n_k} R_{k_v,\eta_{k_v}^{(t)}} \tag{10}$$

This sum-rate will be used as a measure of network performance. All the above parameters are defined as a function of the learning association vector $\boldsymbol{\eta}^{(t)}$ for each learning step t.

Based on the above definitions, then the best-to-date association vector for transmission up to time step T can be defined as

$$\boldsymbol{\beta} = \underset{t=\{1,\ldots,T\}}{\arg\max}\ r(\boldsymbol{\eta}^{(t)}) \tag{11}$$

In a similar way as in (8)-(10), we can define the data rates for transmissions as a function of the best-to-date association vector $\boldsymbol{\beta}$.

D. Optimization Problem

In user association problem, the ultimate goal is to find an optimal association vector which maximizes a network utility function. In this paper, we consider the widely-used sum-rate utility function defined in (10). Thus, the user association optimization problem at learning time step t can be written as

$$\underset{\boldsymbol{\eta}^{(t)}}{\text{maximize}} \quad r(\boldsymbol{\eta}^{(t)}) \tag{12a}$$

$$\text{subject to} \quad \sum_{j\in\mathcal{J}} \mathbb{1}_{k_v,j}^{(t)} \leq 1, \quad \forall k_v \in \mathcal{V} \tag{12b}$$

$$\sum_{k_v\in\mathcal{V}} \mathbb{1}_{k_v,j}^{(t)} \leq D_j, \quad \forall j \in \mathcal{J} \tag{12c}$$

where $\boldsymbol{\eta}^{(t)}$ includes all the (integer) optimization variables $\eta_{k_v}^{(t)}, \forall k \in \mathcal{K}, v \in \{1, 2, \ldots, n_k\}$ (see (5)-(6)). The set of constraints in (12b) represents the unique association constraints, meaning that each VUE can only connect to one BS at each time, and the constraints in (12c) allows our user association scheme to limit each BS's load separately. Assuming the number of data streams (D_j) is chosen based on the available resources at each BS, this set of constraints guarantees that each BS can serve all its associated UEs up to the specified total number of data streams simultaneously.

This optimization problem in its simplest form (without considering the dependency between user association and interference) is known to be NP-hard because of its non-convex nonlinear structure and presence of integer variables [3], [4]. Taking into account this association and interference dependency as formulated in (9) significantly increases the complexity of the problem, and makes exhaustive search the only known technique to guarantee an optimal solution. However, employing exhaustive search quickly becomes infeasible as the network size (number of BSs and UEs) increases. A centralized heuristic approach is proposed in [7], which achieves the optimal solution compared to exhaustive search when it is feasible, but requires full knowledge of CSI between all BSs and UEs at a central entity and suffers from high computational cost and time complexity. Moreover, solving an optimization problem in a highly dynamic network may not be practically feasible, since the optimization is a one-shot for each time slot t, and the problem need to be re-solved at each t as the network dynamics change. In this case, an RL algorithm for user association is more efficient as it can adapt in real time to networks dynamics. While an RL algorithm may not be optimal, it can reach high performance compared to benchmark solutions such as [7]. In this paper, we employ RL and propose efficient centralized and semi-distributed algorithms to achieve a near-optimal solution. We note that while the proposed solutions do not optimally solve this optimization problem, they aim to achieve the highest sum-rate through a learning process while adapting to the changes in the wireless environment and network in real time.

III. Central Load Balancer for Load Balancing Assignment

In a RL-based user association, each VUE takes an action and receives a reward. At learning time step t, each VUE can pick the best BS based on its updated reward vector containing rewards from the connection with each BS. However, due to the load balancing constrains in (7), the decision of each VUE is inter-dependent on the decisions of other VUEs. A collision can happen if the number of VUEs simultaneously picking the same BS is more than its quota allows. In order to avoid collisions, a *load balancing assignment* algorithm is required to determine user associations based on load constraints of all BSs. A load balancing learning algorithm produces a load balanced association vector based on the most recent rewards collected from all UEs. Because of this gathering of information from all UEs, one approach is to enforce these constraints by using a central entity. In this section, we introduce a central entity, the *central load balancer* (CLB), to associate VUEs with BSs according to these load constraints.

A. Upper Confidence Bound Action Selection Rule

We consider a load balancing assignment scheme based on the *UCB* action selection rule, also known as UCB descision-making rule, or UCB policy [18]. This assignment scheme guarantees a balance between exploiting the current best action and exploring other possible actions for each UE. In the UCB action selection approach, at each learning time step t, each VUE k_v wants to be associated with a BS which provides the highest possible reward based on the UCB rule. We define a reward matrix $\boldsymbol{\Omega}$ that contains UCB-updated rewards for all VUEs with the following elements

$$\omega_{k_v,i} = \Gamma_{k_v,i}^{(t-1)} + \sqrt{\frac{2\ln t}{T_{k_v,i}^{(t-1)}}} \tag{13}$$

where $\Gamma_{k_v,j}^{(t-1)}$ is the *updated reward* of VUE k_v received from BS j at the end of learning time step $t-1$, and $T_{k_v,i}^{(t-1)}$ represents the number of times VUE k_v has been associated with BS i up to and including learning time step $t-1$. We also define $\mathbf{T}^{(t)} = [T_{k_v,j}]_{k_v\in\mathcal{V},j\in\mathcal{J}}$ as the *matrix of number of BS selection*. Then the UCB rule specifies user k_v selecting the action (the base station for association) as

$$j = \underset{i\in\mathcal{J}}{\arg\max}\, \boldsymbol{\Omega}_{k_v} \tag{14}$$

where $\boldsymbol{\Omega}_{k_v} = [\omega_{k_v,i}]_{i\in\mathcal{J}}$ is row k_v^{th} of $\boldsymbol{\Omega}$. The UCB technique guarantees a certain and diminishing amount of exploration during the learning process.

If there were no load constraints (quota limits) on the BSs, then each UE k_v can directly implement the resulting choice of (14) as the association decision for the next learning step, independent of other VUEs' choices. With load balancing constraints, however, we need to modify these decisions in order to satisfy the BSs' quotas.

B. Proposed UCB-Based Central Load Balancer

The problem we consider here is an instance of the MA-MAB setting [30], but one with additional constraints that govern the set of actions among participating agents. In traditional MA-MAB problems, multiple agents share the same set of arms and must decide on the same action to take in order to achieve a global goal, which can either be to minimize

Algorithm 1 Joint-UCB Load Balancing Assignment

Input: Learning time step t, reward matrix $\mathbf{\Omega}$, quota vector $\mathbf{q}$, best-to-date sum-rate $r(\boldsymbol{\beta})$

while $\mathbf{\Omega}$ *has nonzero entries* **do**
  J-UCB Action Selection: Select the best VUE-BS pair $[k_v, j]$ according to (15);
  if $q_j > 0$ **then**
    Assign the VUE's learning association: $\eta_{k_v}^{(t)} = j$;
    Update BS's quota: $q_j \leftarrow q_j - 1$;
    Zero out row k_v in reward matrix $\mathbf{\Omega}$;
  else
    Zero out column j in reward matrix $\mathbf{\Omega}$;
  end
end
Calculate sum-rate $r(\boldsymbol{\eta}^{(t)})$ according to (10);
if $r(\boldsymbol{\eta}^{(t)}) > r(\boldsymbol{\beta})$ **then**
  Update the best-to-date association vector $\boldsymbol{\beta} = \boldsymbol{\eta}^{(t)}$;
end

Output: Learning association vector $\boldsymbol{\eta}^{(t)}$ and best-to-date association vector $\boldsymbol{\beta}$

the total regret [31]–[33] or to maximize the total reward [34]. The agents achieve the goal often by communicating their actions with each other in a pair-wise fashion, over which a action-decision rule such as the majority rule is used to select the common action which would converge over time. In these traditional MA-MAB problems, however, the actions of these agents are not explicitly constrained by any condition. In our considered problem, however, the load balancing at the base stations place explicit constraints on the actions of any group of agents at any one time, such that no more than a certain number of agents can take the same action to connect to a certain base station. Furthermore, traditional MA-MAB problems lead to the same final action for all agents, whereas our problem does not require the agents to take the same action in the end. On the contrary, the agents here are competing in the sense that they can take different actions to maximize their own reward. Only that their actions are constrained explicitly by the load balancing conditions.

Specifically, due to the load balancing constrains in (7), the decision of each VUE is inter-dependent on the decisions of other VUEs. Here, we introduce a *joint-UCB (J-UCB) action selection* mechanism to address this dependency. Based on the J-UCB, we then propose a centralized entity (algorithm), the CLB, to collect the reward vectors of all UEs and perform load balancing assignment algorithm. The load balancing assignment algorithm is performed at the CLB which has the knowledge of the entire updated reward matrix $\mathbf{\Omega}$ and the history matrix $\mathbf{T}$ at the beginning of learning time step t.

The assignment algorithm repeatedly performs the following two steps:

1) *J-UCB Action Selection*: Select the VUE-BS pair with the maximum reward in the current $\mathbf{\Omega}$ matrix as

$$[k_v, j] = \underset{l_u \in \mathcal{V}, i \in \mathcal{J}}{\arg\max}\, \mathbf{\Omega} \tag{15}$$

where $\mathbf{\Omega} = [\omega_{l_u,i}]_{l_u \in \mathcal{V}, i \in \mathcal{J}}$ represents the input reward matrix.

2) *Updating* $\mathbf{\Omega}$: Update the reward matrix by zeroing out row k_v of $\mathbf{\Omega}$, and zeroing out column j if the quota of BS j is full, to form a new $\mathbf{\Omega}$.

These two steps are repeated until associations are identified for all VUEs. That is, an association occurs according to (15) by selecting the VUE-BS connection with the highest reward. After an association happens, the association vector $\boldsymbol{\eta}^{(t)}$ is updated, the corresponding row from $\mathbf{\Omega}$ is zeroed out, and the quota of serving BS is updated. If a BS runs out of quota, we zero out the corresponding column from $\mathbf{\Omega}$. These steps are repeated until all entries of reward matrix are equal to zero, i.e., $\mathbf{\Omega} = \mathbf{0}$. At this point, the load-balanced vector $\boldsymbol{\eta}^{(t)}$ is complete and specifies the associations of all VUEs. The J-UCB load balancing assignment is formally described in Alg. 1.

Lemma 1: The proposed J-UCB load balancing assignment in Alg. 1 ensures no collisions among all UEs' actions while simultaneously satisfying all BSs' load-balancing constraints.
Proof: The zeroing out of row k_v after user k_v is selected ensures unique association, such that each VUE is connected to only a single BS. The zeroing out of column j if the quota of BS j is full ensures no collisions of more VUEs connecting to a BS than its quota. At the end of this process, all BSs quota or load constraints are observed. ■

IV. MAB Algorithms for User Association in Static Networks

In this section, we study the user association problem in a static network, where the network settings, including wireless channels and user locations, are static for a duration of T learning time steps (as in a block fading channel). We propose an MAB-based learning algorithm and refer to it as the *basic learning* (BL) algorithm, as a basis for later algorithms dealing with mobility and channel dynamics. The goal is to adapt the association vector $\boldsymbol{\beta}$ for data transmission which specifies the best-to-date connections between BSs and VUEs based on all association vectors $\boldsymbol{\eta}^{(t)}$ learned up to the current time t.

We cast our user association problem as a stateless MA-MAB model where VUEs (agents) are not associated with any state, and their received rewards from BSs (arms) cannot be attributed to any specific distribution [35]. We start with defining the components of our proposed MAB-based learning approach and discuss its regret analysis and convergence prop. Then, we describe the reward updating rule for BL. The location at which this update occurs in the system, at the UE or a BS, will require different signaling mechanisms and timing phases, which leads to different learning algorithms. We use the BL updating rule to design two algorithms: a centralized one where update occurs at a BS, and a semi-distributed one where update is carried out by the UE. We then discuss in detail the signaling overhead and computational complexity of these two algorithms.

A. MAB Components

In this subsection, we define the main components of our proposed MAB-based user association framework.

Algorithm 2 CLB Updating Rule for Basic Learning (BL)

Input: α, $\mathbf{\Gamma}^{(t-1)}$, $\mathbf{T}^{(t-1)}$, $R^{(t)}_{k_v,j}$ for all $k_v \in \mathcal{V}$
for $k_v \in \mathcal{V}$ **do**
  - Update $\Gamma^{(t)}_{k_v,j} = \Gamma^{(t-1)}_{k_v,j} + \alpha(R^{(t)}_{k_v,j} - \Gamma^{(t-1)}_{k_v,j})$;
  - Update $T^{(t)}_{k_v,j} = T^{(t-1)}_{k_v,j} + 1$;
end
Output: Updated $\mathbf{\Gamma}^{(t)}$

1) Environment: The *environment* is the cellular network composed of all BSs, their locations with respect to UEs, and all channels between each UE-BS pair.

2) Agent: Each VUE is an *agent* which interacts with the environment to achieve a goal. As a result, the proposed user association framework is an MA-MAB with $|\mathcal{V}|$ agents. At each learning time step, each agent takes an action according to a policy and receives an instantaneous reward.

3) Action: At each learning time step, VUE (agent) k_v takes an *action* $a^{(t)}_{k_v} \in \mathcal{J}$ indicating that VUE k_v selects (or wishes to select) BS $a^{(t)}_{k_v}$ for association. The current action at learning step t will be the next association at learning time step $t+1$, i.e., $\eta^{(t+1)}_{k_v} = a^{(t)}_{k_v}$.

4) Policy: The solution of a bandit problem is a *policy* (action selection rule) that determines which action should be taken at each learning step. In this paper, we employ the well-known UCB policy as defined in Sec. III-A [18]. Thus, the policy of our proposed MA-MAB algorithm $\pi^{(t)}$ at learning step t is a $|\mathcal{V}| \times 1$ vector defined as: $\boldsymbol{\pi}^{(t)} \triangleq [\boldsymbol{\pi}^{(t)}_1, \ldots, \boldsymbol{\pi}^{(t)}_K]^T$, where $\boldsymbol{\pi}^{(t)}_k \triangleq [a^{(t)}_{k_1}, \ldots, a^{(t)}_{k_{n_k}}]^T$ represents the policy vector of UE k.

5) Instantaneous Reward: Each VUE receives an *instantaneous reward* $R^{(t)}_{k_v,j}$ after taking an action at each learning time step. This instantaneous reward will be used to update the reward value of VUE k_v ($\Gamma^{(t)}_{k_v,j}$) according to an updating rule (see (17), (24), (25)). The ultimate goal of each VUE is to make better decisions during the learning process to achieve a higher expected reward.

6) Reward Vector: Each VUE k_v maintains and updates a *reward vector* $\mathbf{\Gamma}_k$ which contains the reward for all its actions and has the size of $1 \times J$. While the instantaneous reward indicates the immediate value of selecting a BS, the reward vector is an estimate of the long-term expected reward value of selecting each BS. The (network) reward matrix $\mathbf{\Gamma}$ can be obtained by vertical concatenation of the reward vectors of all VUEs and has the total size of $|\mathcal{V}| \times J$. Our stateless MA-MAB framework requires less data to store compared to other RL methods with states and hence allows this tabular implementation which in turn reduces the complexity of our proposed algorithms and facilitates their online implementations.

B. Regret and Convergence

Cumulative regret is the most common metric to measure the performance of an RL policy and is defined as the amount of loss due to deviating from the optimal strategy. For each agent k_v, the cumulative regret (*external regret*) of any RL policy for stateless bandits up to learning step T is defined as [35]

$$\mathfrak{R}_{k_v} = \max_{j \in \mathcal{J}} \mathbb{E}\left[\sum_{t=1}^{T} R^{(t)}_{k_v,j}\right] - \mathbb{E}\left[\sum_{t=1}^{T} R^{(t)}_{k_v,a^{(t)}_{k_v}}\right] \tag{16}$$

This external regret can be bounded above by another notion of regret for adversarial bandits, called *internal regret*, as $\mathfrak{R}_{k_v} \leq J.\mathfrak{R}^{\text{Int}}_{k_v}$ [36]. From each agent's perspective, an MA-MAB model can be seen as a game with two players: the agent itself, and the set of other agents. For internal regret, the following theorem regarding convergence holds:

Theorem 1 ([36], [37]): In an MA-MAB game with a set of agents $\mathcal{V}$, if each agent $k_v \in \mathcal{V}$ plays according to some policy that exhibits per-round vanishing internal regret, i.e., $\lim_{T\to\infty} \frac{1}{T}\mathfrak{R}^{\text{Int}}_{k_v} = 0$, the game converges to the set of correlated equilibrium in a time-average sense.

It is shown that UCB policy guarantees an expected regret of $O(\log T)$ [38], and thus satisfies the vanishing condition in Theorem 1, meaning that our proposed UCB-based MA-MAB algorithms converge to an equilibrium. Our simulation results in Sec. VII-A show that this convergence is quite fast and occurs in only few number of learning steps.

C. Updating Rule for Basic Learning (BL)

In the proposed learning framework, the reward matrix $\mathbf{\Gamma}$ and the matrix of number of BS selection $\mathbf{T}$ are updated at each learning time step t. Each VUE k_v takes an action (selects BS $j = \eta^{(t)}_{k_v}$) at learning time step t and receives an instantaneous reward $R^{(t)}_{k_v,j}$. The expected reward of VUE k_v is then updated as follows [18]

$$\Gamma^{(t)}_{k_v,j} = \Gamma^{(t-1)}_{k_v,j} + \alpha\Big(R_{k_v,j} - \Gamma^{(t-1)}_{k_v,j}\Big) \tag{17}$$

where α is the learning rate. The number of BS selection of VUE k_v is also updated as $T^{(t)}_{k_v,j} = T^{(t-1)}_{k_v,j} + 1$. Since the network is static and there is no handover, we call these steps the updating rule for basic learning (BL) as shown in Alg. 2.

D. Centralized MAB User Association Algorithm

Here, we propose a centralized user association algorithm in which the reward update (17) occurs at the CLB. In each learning time step $t \leq T$, the CLB executes the UCB load balancing assignment (Alg. 1) to obtain association vector $\boldsymbol{\eta}^{(t)}$ and informs all UEs of their connections for the next learning time step. It also compares the new network sum-rate resulting from $\boldsymbol{\eta}^{(t)}$ with the current best-to-date value. If the new sum-rate is higher, the CLB updates the best-to-date association vector as $\boldsymbol{\beta} = \boldsymbol{\eta}^{(t)}$, and informs all UEs of this new $\boldsymbol{\beta}$ for data transmission.

The process at each VUE k_v takes place in five phases. During the first phase, the VUE receives $\eta^{(t)}_{k_v}$ to be used for learning, and the best-to-date association β_{k_v}. Then, in the second and third phase, the VUE connects to its assigned-for-learning-purposes BS $j = \eta^{(t)}_{k_v}$, measures an instantaneous reward $R^{(t)}_{k_v,j}$ and reports it to the CLB. The VUE implements the best-to-date association $j^\star = \beta_{k_v}$ in phase four and maintains it for data transmission in the fifth phase (see Fig. 1.a). After receiving the $R^{(t)}_{k_v,j}$ in phase three, the

Algorithm 3 Centralized MAB Load Balancing User Association

Input: Learning rates α, BSs' quota vector $\mathbf{q}$, initial reward matrix $\mathbf{\Gamma}^{(0)}$, initial matrix of number of BS selection $\mathbf{T}^{(0)} = \mathbf{0}$

for $t = 1 : T$ **do**
- **Central load-balancer (CLB)**:
- - Applies UCB formula to obtain input reward matrix $\mathbf{\Omega} = \mathbf{\Gamma}^{(t-1)} + \sqrt{\frac{2\ln(t)}{\mathbf{T}^{(t-1)}}}$;
- - Executes Alg. 1 to obtain $\boldsymbol{\eta}^{(t)}$ and $\boldsymbol{\beta}$;
- - Informs VUEs of their $\eta_{k_v}^{(t)}$, and β_{k_v} if changed;
- **Each VUE** k_v:
- - Connects to BS $j = \eta_{k_v}^{(t)}$;
- - Receives reward $R_{k_v,j}^{(t)}$ and reports it to CLB;
- **CLB**:
- - Executes an updating rule: **BL** (Alg. 2), or **RTL** (Alg. 5), or **BGL** (Alg. 6) to obtain $\mathbf{\Gamma}^{(t)}$ and $\mathbf{T}^{(t)}$;

end

Output: Best-to-date association vector $\boldsymbol{\beta}$ (up to time step T), $\mathbf{\Gamma}^{(T)}$ and $\mathbf{T}^{(T)}$

Algorithm 4 Semi-Distributed MAB Load Balancing User Association

Input: Learning rate α, BSs' quota vector $\mathbf{q}$, initial reward matrix $\mathbf{\Gamma}^{(0)}$, initial matrix of number of BS selection $\mathbf{T}^{(0)} = \mathbf{0}$

for $t = 1 : T$ **do**
- $\mathbf{q}^{\text{temp}} = \mathbf{q}$;
- **Each VUE** k_v:
- - Applies to best BS according to (14): $j = \arg\max_{i \in \mathcal{J}} \left(\Gamma_{k_v,i}^{(t-1)} + \sqrt{\frac{2\ln t}{T_{k_v,i}^{(t-1)}}} \right)$;
- - **if** $q_j^{temp} > 0$ **then**
- - - VUE k_v receives new reward $R_{k_v,j}^{(t)}$ from BS j;
- - - BS j updates its quota: $q_j^{\text{temp}} \leftarrow q_j^{\text{temp}} - 1$;
- - **else**
- - - BS j rejects VUE k_v;
- - - VUE k_v receives new reward $R_{k_v,j}^{(t)} = 0$
- - **end**
- - Executes an updating rule: **BL** (Alg. 2), or **RTL** (Alg. 5), or **BGL** (Alg. 6) to obtain $\mathbf{\Gamma}_{k_v}^{(t)}$ and $\mathbf{T}_{k_v}^{(t)}$;
- - Reports $\mathbf{\Gamma}_{k_v}^{(t)}$ to CLB;
- **Central load-balancer (CLB)**
- - Executes Alg. 1 with $\mathbf{\Omega} = \mathbf{\Gamma}^{(t)}$ to obtain $\boldsymbol{\beta}$;
- - Informs VUEs about their β_{k_v} if changed;

end

Output: Best association vector $\boldsymbol{\beta}$ (up to time step T), $\mathbf{\Gamma}^{(T)}$ and $\mathbf{T}^{(T)}$

CLB execute the updating rule in (17) to obtain the reward matrix $\mathbf{\Gamma}^{(t)}$ and the matrix of number of BS selection $\mathbf{T}^{(t)}$. In this algorithm, the first three phases are dedicated for learning which use current association result (instead of the best-to-date) to allow sufficient learning exploration, whereas the last two phases are for actual data transmission which use the best-to-date association in order to achieve the highest data rate. A summary of this centralized algorithm is shown in Alg. 3. This algorithm is centralized in the sense that the CLB makes all association decisions for learning and transmission phases.

E. Semi-Distributed MAB User Association Algorithm

Next, we introduce a semi-distributed algorithm in which each VUE performs the reward updating in (14) and proposes an association decision based on its local reward history. Distributed approaches provide low-complexity solutions with minimal signaling overhead between network entities. Distributed algorithms performance, however, is usually worse than that of centralized algorithms since association decisions for learning purposes are made based on local and not global information.

For load balancing user association, the difficulty in implementing a fully-distributed algorithm comes from the fact that the association decision of each individual UE based on their local information does not guarantee load balancing. This drawback is due to the lack of information about the association of other UEs. While each UE can perform its learning procedure in a distributed fashion, but we still need a central entity to track the association of all UEs and enforce the load balancing constraints. This idea leads us to a semi-distributed MAB algorithm as follows.

In a semi-distributed algorithm, instead of receiving an action from the CLB, each VUE proposes an action based on its locally updated reward vectors. At each learning time step t, each VUE follows a six-phase operation: (i) applying to a BS for learning, (ii) receiving the associated instantaneous reward, (iii) reporting updated reward, (iv) receiving best-to-date association β_{k_v} from CLB, (v) performing association for transmission, and (vi) carrying out data transmission (see Fig. 1.b). In particular, each VUE k_v uses the UCB formula in (14) to find best BS providing highest reward. Then, the VUE executes an *apply-response mechanism*, in which the VUE applies to its best BS and receives an instantaneous reward. The reward will be a positive value if the BS has enough quota, but will be zero if the BS is fully loaded. Based on this instantaneous reward, the VUE updates its local reward value according to (17) and also the number of times it has applied to that BS. Then, each VUE k_v reports its $\Gamma_{k_v,j}$ and $T_{k_v,j}$ to the CLB.

In this algorithm, similar to Alg. 3, the CLB is responsible for balancing the loads of BSs by performing the Alg. 1 using the updated rewards, keeping track of the best-to-date association $\boldsymbol{\beta}$, and informing VUEs about this $\boldsymbol{\beta}$ for data transmission. This algorithm is semi-distributed in the sense that each VUE updates its expected reward based on its own decision, instead of the CLB updating rewards as in Alg. 3. This semi-distributed algorithm is summarized in Alg. 4.

F. Signaling Overhead and Complexity Analysis

1) Signaling Overhead: In both the proposed algorithms, information exchange only happens between UEs and the CLB, and there is no communication or exchange among UEs.

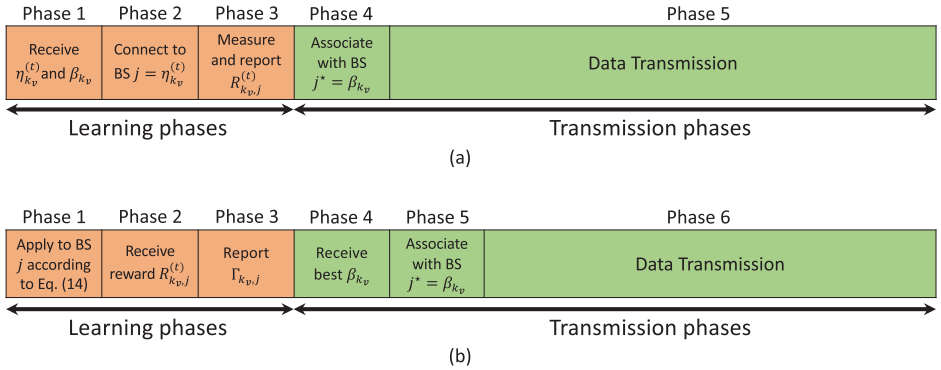

Fig. 1. Learning and transmission phases of VUE k_v in a) Centralized MAB algorithm, and b) Semi-distributed MAB algorithm, during a single learning time step t.

Hence the overhead here refers to the amount of signaling exchange between UEs and the CLB. We first analyzed the signaling overhead of the centralized MAB algorithm. Assume that reporting an instantaneous reward $R_{k_v,j}$ of a VUE to CLB requires m_1 bits. Since each UE k reports n_k rewards, the total number of bits sent to CLB is $Q_1 = \sum_{k=1}^{K} \sum_{v=1}^{n_k} m_1$. After performing the load balancing assignment (Alg. 1), the CLB needs to inform VUEs about their associations for both learning ($\boldsymbol{\eta}^{(t)}$) and transmission (only if $\boldsymbol{\beta}$ changes) phases. Assuming each signaling of an association variable requires m_2 bits, the total number of bits for each association vector is equal to $Q_2 = \sum_{k=1}^{K} \sum_{v=1}^{n_k} m_2$. This process repeats at each learning time step. Thus, the overall signaling overhead per learning time step for this centralized MAB user association is $Q^{\text{cent}} = Q_1 + 2Q_2$ bits.

In the semi-distributed algorithm, the reward reporting from VUEs to the CLB is similar to the centralized one and requires Q_1 bits. After performing the load balancing assignment in Alg. 1, the CLB is only required to inform UEs about their best associations (only if $\boldsymbol{\beta}$ changes), which necessitates Q_2 bits of signaling. Thus, the overall signaling overhead per learning time step for this semi-distributed MAB user association is $Q^{\text{semi-dist}} = Q_1 + Q_2$ bits, lower than the centralized algorithm.

The reward reporting mechanism in the proposed algorithms is equivalent to measurement reporting in the max-SINR scheme, for which informing UEs about their associations necessitates Q_2 bits of signaling, resulting in overall signaling overhead of $Q^{\text{max-SINR}} = Q_1 + Q_2$ bits. Thus, max-SINR incurs the same signaling overhead as the semi-distributed algorithm, which is slightly lower than for the centralized algorithm.

As shown in Fig. 1, the semi-distributed algorithm has more phases which results in a longer time overhead compared to the centralized algorithm. In particular, after Phase 3 in the semi-distributed algorithm, the UE needs to wait for the CLB to produce and send the new association for transmission (β_{k_v}) if it changes. In the centralized approach, however, there is no waiting time since the VUE receives its association for transmission in Phase 1 and can use it immediately in transmission phases (Phases 4-5). We note, however, that the learning pace of the semi-distributed algorithm is faster than that for the centralized algorithm, as shown later in learning convergence results. The semi-distributed algorithm converges faster, albeit to a lower reward and performance value.

2) Computational Complexity: Given that computing instantaneous rewards and updating the expected reward matrix ($\boldsymbol{\Gamma}^{(t)}$) and number of BS selection ($\mathbf{T}^{(t)}$) are simple scalar multiplication and addition operations, the computation complexity of both the centralized and semi-distributed MAB algorithms is dominated by executing Alg. 1. The cost of sorting algorithm, finding "$\arg\max$" over a set of n variables, is $\mathcal{O}(nlog(n))$ [39]. During each *while* loop in Alg. 1, the "$\arg\max$" is taken over all the nonzero elements of the reward matrix $\boldsymbol{\Omega}$. Thus, the complexity of each loop is $\mathcal{O}(LJ\log(LJ))$. Since the total number of loops is in order of L, the total cost of executing Alg. 1 is $\mathcal{O}(L^2 J\log(LJ))$.

In short, the semi-distributed algorithm requires less signaling for communications between the UEs and BSs. For complexity, computation costs mainly incur at the CLB which leads to a similar computation complexity cost for both algorithms.

V. Mobility and Measurement Models

We now consider a dynamic network with user mobility and wireless channel variation, and extend our learning algorithms for both user association and handover. In this section, we discuss mobility model, measurement model, and performance metrics for handover.

A. Mobility Model

We consider UE movement across the network according to the *modified random waypoint* (MRWP) model, which is close to the Levy walk model and human mobility patterns [40]. This model is defined by an infinite sequence of quadruples as $\{(\mathbf{X}_{k,n-1}, \mathbf{X}_{k,n}, V_{k,n}, S_{k,n})\}_{k\in\mathcal{K},\ n\in\mathbb{N}}$, where n denotes the nth *moving step* during which UE k travels from starting waypoint $\mathbf{X}_{k,n-1}$ with coordinate on a 2D plane to a target waypoint $\mathbf{X}_{k,n}$. $V_{k,n}$ represents the random velocity which is uniformly chosen in the range $(0, V_{\max}]$, and $S_{k,n}$ is the random pause time at the target waypoint.

In this mobility model, given a source waypoint $\mathbf{X}_{k,n-1}$, a homogeneous Poisson point process (PPP) $\Phi(n)$ with intensity λ is generated, and the nearest point in $\Phi(n)$ is selected as the target waypoint, i.e., $\mathbf{X}_{k,n} = \arg\min_{\mathbf{x}\in\Phi(n)} ||\mathbf{x} - \mathbf{X}_{k,n-1}||$.

Thus, the *transition length* of UE k during moving step n can be calculated as $L_{k,n} = ||\mathbf{X}_{k,n} - \mathbf{X}_{k,n-1}||$, and its *transition time* is $T_{k,n} = L_{k,n}/V_{k,n}$.

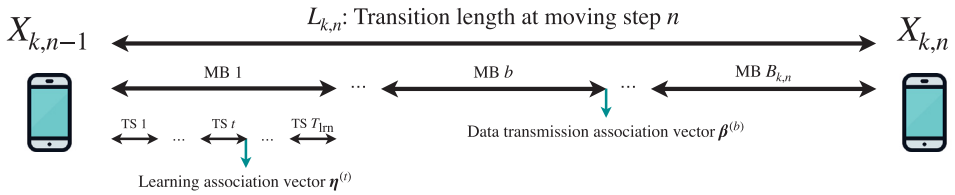

Fig. 2. Structure of moving step n during which UE k travels from source waypoint $\mathbf{X}_{k,n-1}$ to target waypoint $\mathbf{X}_{k,n}$ with velocity $V_{k,n}$. The number of MBs for each moving step is obtained according to (18), and depends on UE velocity $V_{k,n}$, the distance between source and target waypoints $L_{k,n}$, and the time duration of each MB T^{MB}. The learning process is fulfilled inside each MB based on T measurements, each obtained in a learning time step t.

We note that our proposed online user association algorithm in the next section is not specific to the MRWP model and can be applicable to other mobility models, as long as the model provides a way to compute the transition length and time.

B. Measurement Model

Each moving step includes multiple *measurement blocks*[1] (MBs), each with a time duration T^{MB}. Inside each MB, each UE can perform multiple measurements from its serving and neighbor BSs. According to 3GPP standards [42], for each measurement, UE suspends its communication with the serving BS (for a duration called measurement gap) to measure a neighbor BS. This mechanism helps the UE to engage in data transmission while effectively measuring the neighbor cells for the learning purposes. The measurement quantities, as currently specified in the 3GPP standards, can be the reference signal received power (RSRP), reference signal received quality (RSRQ), or SINR, with two measurement reporting options: periodic and event-based [41]. These measurements are then reported to the network every T^{MB} seconds for the purpose of handover. Thus, the number of MBs for UE k during moving step n is determined by rounding up the ratio between the transition time of the UE and the report interval as follows:

$$B_{k,n} = \left\lceil \frac{T_{k,n}}{T^{\text{MB}}} \right\rceil \tag{18}$$

As a result, $B_{k,n}$ is a random number and changes from one moving step to another. At the end of each MB, the network can decide about handover decisions based on the new measurements, which may lead to changes in associations.

Each MB contains a fixed number of measurements denoted as T which corresponds to the total number of learning steps in static network case. We consider a block timing model in which each new measurement inside a MB corresponds to a learning time step t. At the end of each learning time step t, a new learning association vector $\boldsymbol{\eta}^{(t)}$ is produced, and at the end of each MB b, the best-to-date association vector $\boldsymbol{\beta}^{(b)}$ is declared which will be used for data transmission in the next MB $b+1$. We note that it is also possible to update the best-to-date association at the end of each learning time step, but it may result in less stable associations and frequent handover problem. At each learning step t, the CLB specifies from which BS each VUE should collect the measurement. Each MB b has a best-to-date association vector $\boldsymbol{\beta}^{(b)}$, chosen from the best-to-date association vector $\boldsymbol{\beta}^{(b-1)}$ and all the learning association vector $\boldsymbol{\eta}^{(t)}$ within that MB as

$$\boldsymbol{\beta}^{(b)} = \underset{t=\{0,\ldots,T\}}{\arg\max}\ r(\boldsymbol{\eta}^{(t)}) \tag{19}$$

where $r(.)$ is given in (10) and the initial learning association vector at each MB b is the best-to-date association vector from the previous MB $b-1$, that is $\boldsymbol{\eta}^{(0)} = \boldsymbol{\beta}^{(b-1)}$. In this way, $\boldsymbol{\beta}^{(b)}$ is the best-to-date association vector taking into account all past history. Fig. 2 shows the block timing model of a typical UE k traveling from starting waypoint $\mathbf{X}_{k,n-1}$ to target waypoint $\mathbf{X}_{k,n}$ with velocity $V_{n,k}$.

C. Handover Performance Metrics

We define important handover metrics in cellular network, including *handover rate* and *sojourn time*. Considering $\boldsymbol{\beta}^{(b)}$ as the best-to-date association vector by the end of MB b, then $\boldsymbol{\beta}^{(b)} = \boldsymbol{\beta}^{(b+1)}$ means no handover occurred from MB b to MB $b+1$, and $\boldsymbol{\beta}^{(b)} \neq \boldsymbol{\beta}^{(b+1)}$ indicates that there is at least one handover.

1) Handover Rate: In order to calculate the handover rate, we first define $Z_{k_v,n}$ as the number of handovers for VUE k_v during moving step n, which can be calculated by comparing the best-to-date association vectors (e.g. $\boldsymbol{\beta}^{(b)}$ and $\boldsymbol{\beta}^{(b+1)}$, $b \in \{1, 2, \ldots, B_{k,n}\}$) in consecutive MBs as follows:

$$Z_{k_v,n} = \sum_{b=1}^{B_{k,n}} \bar{\delta}\left(\beta_{k_v}^{(b)}, \beta_{k_v}^{(b+1)}\right) \tag{20}$$

The average handover rate per VUE during N moving steps can be then obtained as

$$\overline{\mathcal{R}} = \frac{\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{k_v=1}^{n_k} Z_{k_v,n}}{|\mathcal{V}|\sum_{n=1}^{N} T_n} \tag{21}$$

where $|\mathcal{V}|$ represents the total number of VUEs.

2) Sojourn Time: The sojourn time is the time duration during which the VUE k_v remains associated with a serving BS. Since handover can happen at the end of each MB, the sojourn time is a multiple of MB time duration T^{MB} defined by

$$\tau_{k_v}^{(b)} = N_{k_v}^{(b)}.T^{\text{MB}} \tag{22}$$

where $N_{k_v}^{(\text{b})}$ is the number of MBs during which the association of VUE k_v has remained unchanged, which can be

[1]This parameter is known as information element *ReportInterval* in 3GPP standardization, which indicates the interval between periodical reports. The range for the ReportInterval in 5G NR is 120 ms - 60 min [41].

calculated as

$$N_{k_v}^{(\mathrm{b})} = \sum_{i=0}^{b-1} \left[\delta\left(\beta_{k_v}^{(i)}, \beta_{k_v}^{(i+1)}\right) \prod_{j=i}^{b-1} \delta\left(\beta_{k_v}^{(j)}, \beta_{k_v}^{(j+1)}\right) \right]. \quad (23)$$

In Sec. VII, we employ these handover metrics to evaluate the performance of our proposed learning-based user association algorithms.

VI. MAB Algorithms for User Association and Handover in Dynamic Networks

In this section, we extend the basic MAB learning algorithms to design association and handover algorithms for a dynamic network under user mobility with handover costs. The proposed algorithms can be implemented online and can track network dynamics including user mobility and wireless channel variations at both small and large scales. The algorithms utilize all available measurements at each MB and performs an MAB user association per MB, such that at the end of each MB b, each algorithm produces a best-to-date association vector $\boldsymbol{\beta}^{(b)}$. Specifically, we adapt the reward updating rules to include new measurements and handover costs. These updating rules can then be used in either the centralized or semi-distributed MAB algorithm to make these algorithms adapt to user mobility and channel dynamics.

A. Updating Rules With Handover Cost

In order to integrate the effect of user mobility and channel dynamics in the learning process, we apply a handover cost in the learning process by considering a reduction in the instantaneous reward. Since there are two sets of association vectors, one for learning ($\boldsymbol{\eta}$) and one for transmission ($\boldsymbol{\beta}$), we propose two different rules for reward updating in the learning process.

The first option is to apply a cost on the instantaneous reward if the current learning association of VUE k_v differs from its best-to-date association from the previous MB (i.e., $\eta_{k_v}^{(t)} \neq \beta_{k_v}^{(b-1)}$). In this case, we define the *best-to-date updating rule (BU)* as follows:

$$\Gamma_{k_v,j}^{(t)} = \Gamma_{k_v,j}^{(t-1)} + \alpha\Big(\big(1 - \zeta(\tau_{k_v}^{(b)}) \bar{\delta}(\beta_{k_v}^{(b-1)}, j)\big) R_{k_v,j} - \Gamma_{k_v,j}^{(t-1)} \Big) \quad (24)$$

where $j = \eta_{k_v}^{(t)}$ represents the current learning association of VUE k_v at learning time step t, $\beta_{k_v}^{(b-1)}$ is the best-to-date association of VUE k_v at the end of previous MB $(b-1)$, and $\zeta(\tau)$ represents the handover cost which is discussed in the next section.

The second option is to apply a handover cost by comparing two consecutive learning associations. That is, the current learning association $j = \eta_{k_v}^{(t)}$ is compared with the previous learning association in the current MB $\eta_{k_v}^{(t-1)}$, and not with $\beta_{k_v}^{(b-1)}$. In this case, we define the *current learning updating rule (LU)* as follows:

$$\Gamma_{k_v,j}^{(t)} = \Gamma_{k_v,j}^{(t-1)} + \alpha\Big(\big(1 - \zeta(\tau_{k_v}^{(b)}) \bar{\delta}(\eta_{k_v}^{(t-1)}, j)\big) R_{k_v,j} - \Gamma_{k_v,j}^{(t-1)} \Big) \quad (25)$$

This updating rule applies a handover cost only if there is an immediate change in the association during the learning process, instead of comparing to $\beta_{k_v}^{(b-1)}$.

A chosen updating rule, either (24) or (25), is applied in the learning process inside the MB b which include multiple learning time steps $t = \{1, 2, \ldots, T\}$. Since the channel dynamics are assumed to be static inside an MB, changes between two consecutive learning associations $\eta_{k_v}^{(t-1)}$ and $\eta_{k_v}^{(t)}$ are most likely caused by user mobility, whereas changes between $\eta_{k_v}^{(t)}$ and β_{k_v} are caused by both user mobility and wireless channel variations. These differences can lead to different probabilities of applying a handover cost, which consequently can lead to trade-offs between the network sum-rate and the handover rate. These trade-offs are analyzed via simulations in Sec. VII.

B. Handover Cost Function

The handover cost function is an important design choice which affects the learning performance. Similar to using two different association vectors, one for transmission and one for learning, here we propose the use of two separate handover costs, one for system performance and one for the learning process. For system performance measures such as user data rate, each handover action incurs a fixed percentage cost due to the handover time overhead that could otherwise be used for data transmission. For the learning process, we propose a variable handover cost as a function of the sojourn time, the time that a VUE has stayed connected with a BS (see (22)). This model includes as a special case the fixed cost model which is prevalent for both learning and performance in the literature. To the best of our knowledge, our proposed concept of using two different cost functions for system performance and learning process is novel.

We consider a *handover cost model for learning* that includes a 'soft' cost component C_d and a 'hard' cost component C_0. The soft-cost C_d can be amortized over time such that the cost decreases the longer an user has stayed connected to the current BS before a handover occurs. The hard-cost component stays fixed throughout the learning process. For example, the hard cost can account for the overhead occurred in a handover process (such as measurements and signaling), and the soft cost can account for the loss in learning performance due to handover causing the rewards to change. Both of these cost components are only for the learning process.

We propose a *learning handover cost* as a decreasing function of the sojourn time as

$$\zeta_{\boldsymbol{\eta}}(\tau) = C_d e^{-\frac{\tau}{10}} + C_0 \quad (26)$$

This learning handover cost is applied in the learning process to produce the learning association vector $\boldsymbol{\eta}^{(t)}$. Here C_d is amortized over time and the handover cost reduces to C_0 as $\tau \to \infty$. Note that the value for C_0 can be zero, which makes the learning cost approach zero asymptotically with increasing sojourn time. This cost function is novel and can lead to higher learning performance by alleviating the frequent handover problem. We note that this decreasing handover cost over longer sojourn time is also simple to implement and is appealing from a practical perspective.

Algorithm 5 CLB Updating Rule for Real-Time Learning (RTL)

Input: Reward matrix $\mathbf{\Gamma}^{(t-1)}$, matrix of number of BS selection $\mathbf{T}^{(t-1)}$

for *each* $k_v \in \mathcal{V}$ *at MB* b **do**
- Perform a reward updating rule with handover cost ((24) or (25));
- Update $T^{(t)}_{k_v,j} = T^{(t-1)}_{k_v,j} + 1$;

end

Output: Updated $\mathbf{\Gamma}^{(t)}$ and $\mathbf{T}^{(t)}$

Algorithm 6 CLB Updating Rule for Background Learning (BGL)

Input: Reward matrix $\mathbf{\Gamma}^{(t-1)}$, matrix of number of BS selection $\mathbf{T}^{(t-1)}$

for $i = 1 : N_{BGL}$ **do**
- Execute Alg. 5 and perform the updates with the same instantaneous reward;

end

Output: Updated $\mathbf{\Gamma}^{(t)}$ and $\mathbf{T}^{(t)}$

In numerical results section, we will also consider two special cases of this learning handover cost model. First is zero cost $\zeta = 0$, in which the updating rule in (24) or (25) reduces to the BL updating rule in (17). This means any handover effect such as reduction in the instantaneous reward due to handover delay is ignored. In a mobile network, this can increase signaling overhead and result in a high handover rate. Second is a fixed cost model, in which $\zeta = C_f$ is constant and does not change over time. This fixed cost model is common in the literature and is simple to obtain a first order effect of handover, but may be sub-optimal in performance since it ignores the effect of sojourn time. Impact on system performance of these two special cases of handover cost will be analyzed and compared with the proposed cost in (26).

In all cases, *the handover cost for system performance* ϵ is a fixed percentage such that the instantaneous rate of VUE k_v resulting from using the best-to-date association vector with handover is

$$\tilde{R}_{k_v,j}(\boldsymbol{\beta}^{(b)}) = (1-\epsilon)\bar{\delta}(\beta^{(b-1)}_{k_v}, j) R_{k_v,j}(\boldsymbol{\beta}^{(b)}) \qquad (27)$$

where $j = \beta^{(b)}_{k_v}$ is the bets-to-date association of VUE k_v. Consequently, the network sum-rate with handover can be calculated according to (10) by using $\tilde{R}_{k_v,j}(\boldsymbol{\beta}^{(b)})$.

C. Proposed User Association and Handover Learning Approaches

Next, we introduce two learning approaches different in the process and the frequency in which the reward updating is carried out. Both approaches utilize the local measurements at UEs to continuously adapt user association to network dynamics, and provide the best-to-date association vector at the end of each MB. During an MB, there are T measurement steps, and during each step, each VUE performs one measurement. The two approaches differ in how the expected rewards are updated during each measurement step.

Both approaches can be applied to either the centralized MAB algorithm (Alg. 3) or the semi-distributed one (Alg. 4) to adapt these algorithms to the dynamic networks. For the purpose of discussion, we will describe these two approaches as applying to the centralized algorithm, where the reward updates are carried out at the CLB. For the semi-distributed algorithm, these updates will be performed at each VUE.

1) Real-Time Learning (RTL): In this approach, at each learning time step t of an MAB algorithm, only a single update is performed according to (24) or (25), based on the new measurement obtained in that learning time step. The association vector obtained at the end of each MB is used as the initial association of the next MB. Because these updates are performed as new measurements arrive, we refer to this approach as *real-time learning* or RTL. The learning process in RTL is updated to match to the same speed as new knowledge of the system is obtained. The CLB updating rule for RTL approach is shown in Alg. 5.

2) Background Learning (BGL): In RL, the reward computed at each step is an estimate of the expected reward that would result if the learning process was carried out for its time duration and reached the final value [18]. Thus this estimate of the reward at each step can be improved by performing the reward update more frequently. One way to do this is to continue to update the expected rewards in-between the measurements. At each time step t of MAB algorithm, each VUE uses its measurement to perform *multiple updates* of its reward vector and vector of number of BS selection. These updates are based on the same measurement at time step t. This approach enhances the effect of most recent measurements compared to the previous ones. The CLB updating rule for this learning approach is given in Alg. 6. In the next section, we provide numerical results to compare these learning approaches which reveal the trade-offs in performance among them.

VII. Numerical Results

We evaluate the performance of the proposed MAB algorithms in the downlink of a mmWave-enabled HetNet, including $J_B = 2$ MBS operating at 1.8 GHz, $J_S = 4$ SBSs operating at 28 GHz, $K = 30$ UEs. The load balancing constraints are specified by BSs' quota vector $\mathbf{q} = [18, 18, 6, 6, 6, 6]$, and we assume each UE requests two data streams ($n_k = 2$), each can be received from either a MBS or a SBS. We pick a fixed value for learning rate $\alpha = 0.8$, which is found via extensive simulations to work well. The channels for sub-6 GHz links and the mmWave links are generated as described in Sec. II-B. We assume each mmWave link is composed of 5 clusters with 10 rays per cluster. In order to implement 3D beamforming, each BS is equipped with a UPA of size 8×8 ($M_j = 64$), and each UE is equipped with a 2-element antenna module designed for sub-6 GHz band, and a 4×1 ULA of antennas designed for mmWave band ($N_k = 4$). The noise power spectral density is -174 dBm/Hz, and the bandwidths for sub-6 GHz band and mmWave band are 20 MHz and 400 MHz, respectively. Also, we assume that the transmit power of MBS is 10dB higher than that for SBSs. Network nodes are deployed in a 500×500 m^2 square

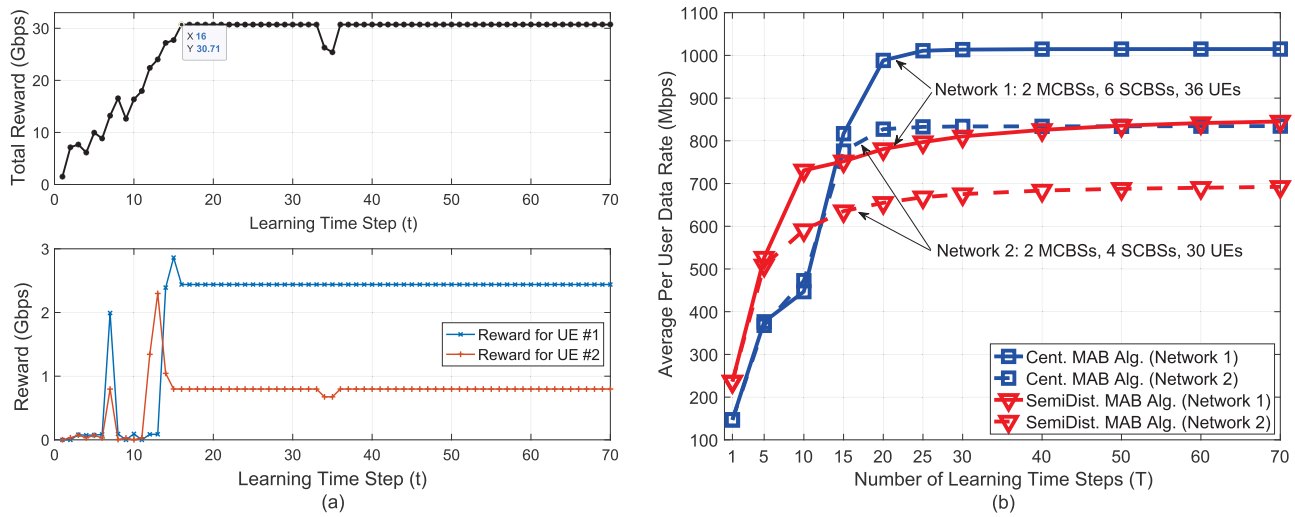

Fig. 3. a) Total expected rewards of all VUEs (upper figure) and expected reward of two typical UEs (lower figure) versus learning time step for the centralized MAB user association algorithm (Network 1). b) Comparing average network sum-rate of the proposed MAB algorithms versus number of learning time steps.

where the BSs are placed at specific locations and the UEs are distributed randomly according to a homogeneous Poisson point process (PPP).

A. Learning Analysis

In this section, we consider a static network and evaluate the convergence of expected rewards and the effect of number of learning steps on network performance.

Fig. 3.a depicts the convergence of the total expected reward of all VUEs and the expected rewards of two typical UEs (including their VUEs) with respect to learning time steps for the centralized MAB algorithm. These figures show a clear trend of that the total reward increasing as learning time step grows. The small number of 16 time steps required for reaching near maximum is encouraging. This result indicates that online implementation of the proposed algorithm can reach close to its best performance even in a reasonable number of learning time steps.

Fig. 3.b depicts the best-to-date per user data rate versus the number of learning steps for the centralized and semi-distributed MAB algorithms for two network sizes. In a wireless network, the number of learning steps correspond to the number of time steps in which the channels stay unchanged, which is set by the channel coherence time. These results show that in both network sizes, the semi-distributed algorithm converges faster than the centralized algorithm, but to a lower data rate value. The convergence for both algorithms, however, is quite fast; for example, the centralized algorithm reaches its maximum average value after only 20-25 steps.

B. Centralized vs. Semi-Distributed MAB Algorithms in a Static Network

Consider a static network with non-CoMP transmission and $t = 50$ learning time steps. Fig. 4 compares the performance of the proposed centralized and semi-distributed MAB user association algorithms with 1) a benchmark as the (non-learning) centralized `WCS` algorithm [7], 2) a centralized `Fractional` association scheme [4], and 3) the

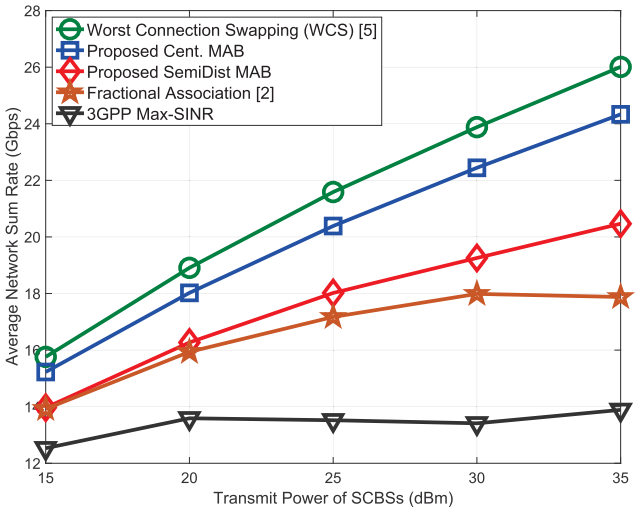

Fig. 4. Comparing average network sum-rate of the proposed MAB algorithms versus transmit power of SBSs. Transmit power of MBSs is 10 dB higher than SBSs.

conventional `3GPP Max-SINR` user association approach. For the `Fractional` scheme we used Alg. 1 to convert fractional associations to integer associations. For the `3GPP Max-SINR` scheme, each UE connects to the BS providing the highest max-SINR, and we also applied Alg. 1 to satisfy the load constraints. This figure shows that both the proposed centralized and semi-distributed MAB algorithms achieve a network throughput close to, within 94-97% and 79-88% of, the `WCS` algorithm, while outperforming both the `Fractional` and `3GPP Max-SINR` algorithms. We note that the proposed learning-based algorithms utilize the available measurements at UEs, and thus have much lower signaling overhead and complexity with respect to `WCS` algorithm which requires full CSI information. This figure also shows that our proposed J-UCB rule used in the centralized algorithm achieves 8-20% higher network sum-rate compared to original UCB rule used in the semi-distributed algorithm.

In Fig. 5, we compare `NonCoMP` and `CoMP` transmission scenarios. In this simulation the quota for MBSs is

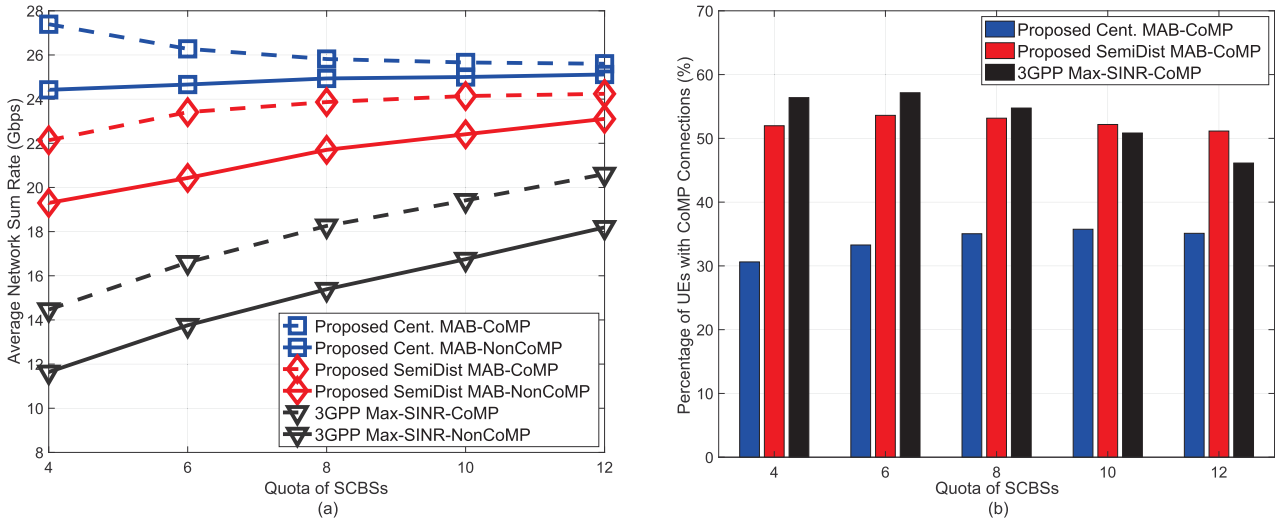

Fig. 5. Comparing a) average network sum-rate, and b) percentage of UEs with CoMP connections, in a network with 6 BSs. Transmit powers of MBSs and SBSs are 45 dBm and 35 dBm, respectively.

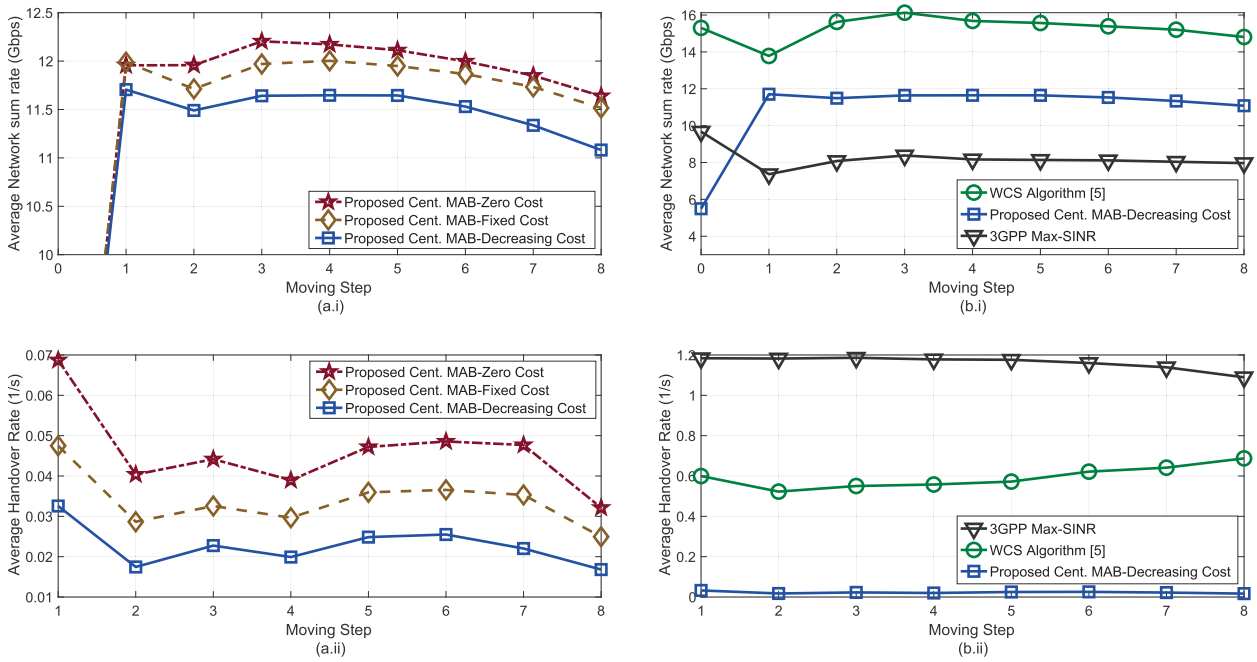

Fig. 6. Comparing average network sum-rate and average handover rate of a) the MAB algorithm with three handover cost models where $C_f = 0.3$, $C_d = 0.9$ and $C_0 = 0$, and b) the MAB algorithm (with decreasing handover cost model) against the WCS and max-SINR algorithms.

fixed ($q_1 = q_2 = 18$), while the quota of SBSs increase from 4 to 12 data streams per BS, and the corresponding number of UEs are $K \in \{26, 30, 34, 38, 42\}$. In the NonCoMP case, each UE is enforced to receive its both data streams ($n_k = 2$) from a single BS, whereas in CoMP transmission, each UE is allowed to connect with multiple BSs and received data streams independently. Subfigure 5.a shows that enabling CoMP results in a higher network sum-rate for all three schemes. For the proposed MAB algorithms the CoMP advantage is more significant when the quota of SBSs and consequently the number of UEs are smaller, because of less interference between data streams. Subfigure 5.b indicates that the proposed centralized algorithm has much lower percentage of CoMP connections compared to max-SINR scheme, which is advantageous because of lower CoMP signal overhead, while still achieving higher network sum-rate.

C. User Association and Handover in a Dynamic Network

We now consider a dynamic network with user mobility and study the handover performance. The handover cost for system performance is $\epsilon = 0.05$. At each moving step $n \in \{1, 2, \ldots, 20\}$, we assume 10 randomly selected (out of $K = 30$) UEs are moving across the network, and each VUE performs $T = 6$ SINR measurements (during each MB) as directed by the CLB. For each moving UE, the next waypoint is generated based on the MRWP model as described in Sec. V-A. The velocity of moving UEs is assumed to be fixed unless otherwise indicated. The number of MBs per moving

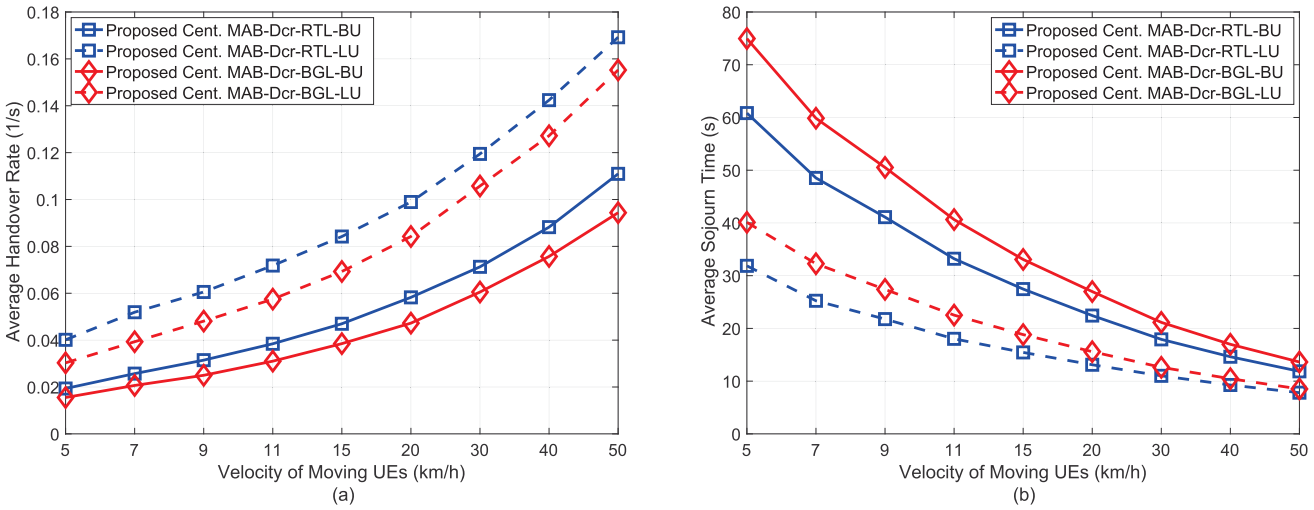

Fig. 7. a) Average handover rate, and b) Average sojourn time of the centralized MAB algorithm with decreasing handover cost ($C_d = 0.9$). The velocity ranges of moving UEs are: 1) walking [5, 7] km/h, 2) running [9, 11] km/h, 3) biking [15, 20] km/h, and 4) city driving [30, 40, 50] km/h.

step is calculated based on (18). For RTL approach (Alg. 5), the reward of each VUE is updated once, while for BGL (Alg. 6), the reward is updated $N_{\text{BGL}} = 10$ times. The 3GPP and WCS algorithms are used as baseline and benchmark comparisons, where the `3GPP` handover is triggered when a neighbor BS becomes an offset amount better than serving BS (Event A3) [41].

Fig. 6 depicts the average network sum-rate and average handover rate of different association schemes versus moving steps. Fig. 6.a shows a clear trade-off between handover rate and achievable data rate among the different learning cost models. The decreasing cost model achieves the lowest handover rate (at 33% lower than fixed cost and 50% lower than zero cost) at only a slight reduction (3-4%) in network sum-rate. As such, we use the decreasing handover cost for all subsequent simulations.

Fig. 6.b compares the performance of the centralized MAB algorithm with decreasing cost model and the `WCS` and `3GPP Max-SINR` algorithms. These results show that the proposed learning algorithm significantly outperforms `3GPP Max-SINR` in terms of both handover rate and network sum-rate, achieving 1.5 times the throughput at an order of magnitude lower in handover rate. Compared to WCS, there is a trade-off between handover rate and network throughput, where the handover rate of the learning algorithm is drastically lower. The initial ramping up of the learning algorithm is due to limited number of measurements and short history at the beginning. As the number of measurements increases and the learning history becomes richer, the proposed algorithm can utilize the learning history and ramps up its performance. This adaptation is quite fast and is fulfilled during the very first few MBs inside the first moving step, attesting to the validity of this learning approach.

Fig. 7 compares `RTL` and `BGL` learning approaches with dynamic updating rules in (24) and (25) indicated by `BU` and `LU`, respectively. For this simulation, we use the Non-CoMP version of centralized MAB algorithm with decreasing handover cost ($C_d = 0.9$, $C_0 = 0$). The results indicate `BU` achieves lower handover rate and higher sojourn time compared to `LU`. This is due to the fact that the best-to-date

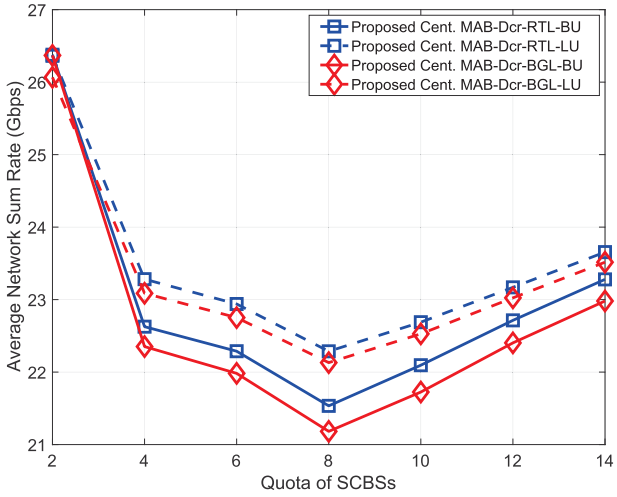

Fig. 8. Comparing average network sum-rate of the RTL and BGL learning approaches with two updating rules given in (24) and (25). Transmit powers of MBSs and SBSs are 45 dBm and 35 dBm, respectively.

association vector β is updated less frequently in `BU`. Moreover, `BGL` outperforms `RTL` in terms of both handover rate and sojourn time. The reason is that `BGL` enhances the effect of most recent measurements by performing multiple reward updating in-between the measurements, thus UEs are more willing to keep their last best-to-date associations.

Fig. 8 compares the `RTL` and `BGL` learning approaches with updating rules `BU` and `LU`. As the total BS quotas and number of UEs grow, the sum-rate decreases at first, due to the fact that BSs are sharing the same amount of power among more UEs. After a threshold point ($q_j = 8,\ j \in \mathcal{J}_S$), however, the sum-rate increases as the higher number of UEs increases the chance of having more UEs closer to SBSs, resulting in higher data rates for those UEs. Compared to the plots in Fig. 7, there is a clear trade-off among the four learning schemes in terms of handover and sum-rate performance.

VIII. Conclusion

Using an MAB reinforcement learning technique, we proposed centralized and semi-distributed learning algorithms

to perform user association and handover in a mmWave-enabled HetNet, while maintaining load balancing. The algorithms explicitly satisfy the load balancing constraints by employing a central load balancer to associate UEs with BSs based on their quotas. We utilized a user mobility model and introduced a measurement model for dynamic networks. We proposed a learning cost model decreasing with connection time to reduce frequent handover, and considered several updating rules for the learning rewards, including real-time and background learning to boost performance. Our numerical results showed that the learning process in these algorithms is fast and efficient, outperforms 3GPP handover scheme, and reaches closely the throughput performance of the benchmark WCS algorithm, while significantly reducing the handover rate. These features make our proposed algorithms potentially suitable for online user association and handover in highly-dynamic mmWave-enabled HetNets.

REFERENCES

[1] A. Alizadeh and M. Vu, "Multi-armed bandit load balancing user association in 5G cellular HetNets," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9348164

[2] S. Ahmadi, *LTE-Advanced: A Practical Systems Approach to Understanding 3GPP LTE Releases 10 and 11 Radio Access Technologies*. Amsterdam, The Netherlands: Elsevier, 2013.

[3] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, Jun. 2013.

[4] D. Bethanabhotla, O. Y. Bursalioglu, H. C. Papadopoulos, and G. Caire, "Optimal user-cell association for massive MIMO wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 1835–1850, Mar. 2016.

[5] G. Athanasiou, P. C. Weeraddana, C. Fischione, and L. Tassiulas, "Optimizing client association for load balancing and fairness in millimeter-wave wireless networks," *IEEE/ACM Trans. Netw.*, vol. 23, no. 3, pp. 836–850, Jun. 2015.

[6] M. Rebato, M. Mezzavilla, S. Rangan, F. Boccardi, and M. Zorzi, "Understanding noise and interference regimes in 5G millimeter-wave cellular networks," in *Proc. 22nd Eur. Wireless Conf.*, May 2016, pp. 1–5.

[7] A. Alizadeh and M. Vu, "Load balancing user association in millimeter wave MIMO networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 6, pp. 2932–2945, Jun. 2019.

[8] P. Han, Z. Zhou, and Z. Wang, "User association for load balance in heterogeneous networks with limited CSI feedback," *IEEE Commun. Lett.*, vol. 24, no. 5, pp. 1095–1099, May 2020.

[9] S. Khosravi, H. S. Ghadikolaei, and M. Petrova, "Learning-based load balancing handover in mobile millimeter wave networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–7.

[10] H. Shokri-Ghadikolaei *et al.*, "Millimeter wave cellular networks: A MAC layer perspective," *IEEE Trans. Commun.*, vol. 63, no. 10, pp. 3437–3458, Oct. 2015.

[11] A. Talukdar, M. Cudak, and A. Ghosh, "Handoff rates for millimeter-wave 5G systems," in *Proc. IEEE 79th Veh. Technol. Conf. (VTC Spring)*, May 2014, pp. 1–5.

[12] M. Chen *et al.*, "Distributed learning in wireless networks: Recent progress and future challenges," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3579–3605, Dec. 2021.

[13] Y. Hu, M. Chen, W. Saad, H. V. Poor, and S. Cui, "Distributed multi-agent meta learning for trajectory design in wireless drone networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3177–3192, Oct. 2021.

[14] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5141–5152, Nov. 2019.

[15] M. Sana, A. De Domenico, and E. C. Strinati, "Multi-agent deep reinforcement learning based user association for dense mmWave networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 6520–6534.

[16] D. Li, H. Zhang, K. Long, W. Huangfu, J. Dong, and A. Nallanathan, "User association and power allocation based on Q-learning in ultra dense heterogeneous networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/9013455

[17] A. Zappone, L. Sanguinetti, and M. Debbah, "User association and load balancing for massive MIMO through deep learning," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Oct. 2018, pp. 1262–1266.

[18] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2015.

[19] S. Kang and C. Joo, "Low-complexity learning for dynamic spectrum access in multi-user multi-channel networks," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2018, pp. 1367–1375.

[20] I. Aykin, B. Akgun, M. Feng, and M. Krunz, "MAMBA: A multi-armed bandit framework for beam tracking in millimeter-wave systems," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Jul. 2020, pp. 1469–1478.

[21] *Study on New Radio Access Technology; Physical Layer Aspects*, document TR 38.802, V. 14.2.0, Sep. 2017.

[22] S. Maghsudi and E. Hossain, "Distributed user association in energy harvesting small cell networks: A probabilistic bandit model," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1549–1563, Mar. 2017.

[23] C. Shen and M. van der Schaar, "A learning approach to frequent handover mitigations in 3GPP mobility protocols," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017, pp. 1–6.

[24] Y. Zhao and X. Luo, "Handover mitigation in dense HetNets via bandit arm elimination," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[25] *Study on Channel Model for Frequencies From 0.5 to 100 GHz*, document TR 38.901, V. 15.0.0, 3GPP, Jun. 2018.

[26] T. A. Thomas, H. C. Nguyen, G. R. MacCartney, and T. S. Rappaport, "3D mmWave channel model proposal," in *Proc. IEEE 80th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2014, pp. 1–6.

[27] M. K. Samimi, T. S. Rappaport, and G. R. MacCartney, Jr., "Probabilistic omnidirectional path loss models for millimeter-wave outdoor communications," *IEEE Wireless Commun. Lett.*, vol. 4, no. 4, pp. 357–360, Aug. 2015.

[28] *E-UTRA; Physical Channels and Modulation*, document Tech. Specification 36.211, V. 12.9.0, 3GPP, Apr. 2017.

[29] *NR; Physical Layer Procedures for Data*, document TS 38.214, v. 15.5.0, Mar. 2018.

[30] L. Busoniu, R. Babuska, and S. B. De, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 2, pp. 156–172, Feb. 2008.

[31] M. Agarwal, V. Aggarwal, and K. Azizzadenesheli, "Multi-agent multi-armed bandits with limited communication," 2021, *arXiv:2102.08462*.

[32] J. Zhu, E. Mulle, C. S. Smith, and J. Liu, "Decentralized multi-armed bandit can outperform classic upper confidence bound," 2021, *arXiv:2111.10933*.

[33] A. Sankararaman, A. Ganesh, and S. Shakkottai, "Social learning in multi agent multi armed bandits," in *Proc. Abstr. Perform. Joint Int. Conf. Meas. Modeling Comput. Syst. (SIGMETRICS)*, Jun. 2020, pp. 1–35.

[34] S. Shahrampour, A. Rakhlin, and A. Jadbabaie, "Multi-armed bandits in multi-agent networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2786–2790.

[35] S. Maghsudi and E. Hossain, "Multi-armed bandits with application to 5G small cells," *IEEE Wireless Commun.*, vol. 23, no. 3, pp. 64–73, Jun. 2016.

[36] S. Maghsudi and S. Stanczak, "Joint channel selection and power control in infrastructureless wireless networks: A multiplayer multiarmed bandit framework," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4565–4578, Oct. 2015.

[37] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[38] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2, pp. 235–256, 2002.

[39] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2009.

[40] X. Lin, R. K. Ganti, P. J. Fleming, and J. G. Andrews, "Towards understanding the fundamentals of mobility in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1686–1698, Apr. 2013.

[41] *5G NR Radio Resource Control (RRC); Protocol Specification*, document TR 38.331, V. 15.7.0, 3GPP, Oct. 2019.

[42] *NR; Requirements for Support of Radio Resource Management*, document TR 38.133, V. 15.13.0, 3GPP, Mar. 2021.

Alireza Alizadeh (Member, IEEE) received the B.Sc. degree in electrical engineering from the Sadjad University of Technology, Mashhad, Iran, the M.Sc. degree in communications engineering from the Ferdowsi University of Mashhad, Mashhad, and the PhD degree in electrical engineering from Tufts University, Medford, MA, USA. In summer 2018, he worked as a Wireless Research and Development Intern at Vivint Wireless, Santa Clara, CA, USA. In summer 2019, he was a Wireless System Test Intern at Apple, Sunnyvale, CA, USA. He is currently a Cellular Protocol Engineer at Apple, Cupertino, CA, USA. His research interests include 5G and beyond 5G cellular networks, 3GPP standardization, mmWave-enabled HetNets, massive MIMO, precoding and beamforming, game theory, and machine learning.

Mai Vu (Senior Member, IEEE) received the bachelor's degree in computer systems engineering from the Royal Melbourne Institute of Technology, Melbourne, VIC, Australia, the M.S.E. degree in electrical engineering from the University of Melbourne, Parkville, VIC, Australia, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA. From 2006 to 2008, she was a Lecturer and a Researcher with the School of Engineering and Applied Sciences, Harvard University, MA, USA. From 2009 to 2012, she was an Assistant Professor of electrical and computer engineering with McGill University, QC, Canada. From 2013 to 2022, she was an Associate Professor of electrical and computer engineering with Tufts University, MA, USA. Since September 2022, she has been a Full Professor of electrical and computer engineering with Tufts University.

Dr. Vu conducts research in wireless systems, signal processing, and networked communications. She has published extensively in the areas of millimeter-wave communications, 5G and 6G systems, cooperative and cognitive communications, MIMO wireless, and energy-efficient communications. She has served on the Technical Program Committee for numerous IEEE conferences. She served as a Co-Organizer for the 2017 New England Workshop on Software Defined Radio (NEWSDR'17) at Tufts University. She is serving as a Technical Co-Chair for the 2023 IEEE Communication Theory Workshop (CTW), Taiwan, and also serving on the AdHoc Committee on Mission Critical Communications, IEEE Vehicular Technology Society. From 2013 to 2016, she was an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.