Transferable and Distributed User Association Policies for 5G and Beyond Networks

Mohamed Sana, Nicola Di Pietro, Emilio Calvanese Strinati

Transferable and Distributed User Association Policies for 5G and Beyond Networks

Mohamed Sana[1], Nicola di Pietro[2], Emilio Calvanese Strinati[1]
[1]CEA-Leti, Université Grenoble Alpes, F-38000 Grenoble, France
[2]Athonet, via Cà del Luogo 6/8, 36050, Bolzano Vicentino (VI), Italy
Email: {mohamed.sana, emilio.calvanese-strinati}@cea.fr, nicola.dipietro@athonet.com

***Abstract*—We study the problem of user association, namely finding the optimal assignment of user equipment to base stations to achieve a targeted network performance. In this paper, we focus on the *knowledge transferability* of association policies. Indeed, traditional non-trivial user association schemes are often scenario-specific or deployment-specific and require a policy re-design or re-learning when the number or the position of the users change. In contrast, transferability allows to apply a single user association policy, devised for a specific scenario, to other distinct user deployments, without needing a substantial re-learning or re-design phase and considerably reducing its computational and management complexity. To achieve transferability, we first cast user association as a multi-agent reinforcement learning problem. Then, based on a neural *attention mechanism* that we specifically conceived for this context, we propose a novel distributed policy network architecture, which is transferable among users with *zero-shot generalization capability* i.e., without requiring additional training. Numerical results show the effectiveness of our solution in terms of overall network communication rate, outperforming centralized benchmarks even when the number of users doubles with respect to the initial training point.**

I. Introduction

With the proliferation of smart connected devices, the cyber and physical spaces are fusing, turning humans, objects and events more and more into exponentially growing sources of digital information [1]. To cope with this, modern wireless networks, such as 5G networks, are becoming denser and heterogeneous with the coexistence of base stations (BSs) operating at different frequencies. In this context, *user association*, namely efficiently finding optimal assignments of user equipments (UEs) to BSs to achieve a targeted network performance, is a crucial challenge because it directly affects the network spectral efficiency and the users' perceived quality of service. In general, for dense networks, this is a challenging task as it involves non-convex and combinatorial optimization problems, whose complexity grows exponentially with the number of UEs. This difficulty is even exacerbated in highly dynamic networks such as in millimeter-wave (mmWave) networks, subject to frequent changes of the radio environment due to highly directional transmissions and variable channel conditions [2]. To address these issues, we propose a scalable and easily manageable user association policy. Our proposed approach is conceived with a specific focus on the key aspect of *transferability*, which allows to apply a user association strategy or policy acquired in a specific scenario (*e.g.* a network deployment) to distinct but related ones, without needing to substantially redesign, recompute or relearn a new policy. This considerably reduces the computational complexity of user association during the network operations and makes the policy adapted to distributed and dynamic scenarios. So far, despite their many appreciable features, solutions of the literature lack transferability. In [3], a distributed user association scheme is proposed using Lagrangian tools. The user association is reformulated as a non-cooperative game with local interactions in [4] and as a matching game in [5]. However, every time the radio environment changes due to, for instance, the arrival or departure of UEs, this solution needs to be recomputed to seek a new convergence point and to correct a possible drift from optimality. Recently, a deep neural network (NN) architecture was introduced in [6] that predicts the user association and power allocation. Similarly, authors in [7] formulated the problem of user association as a multi-label classification problem. In [8], [9], [10], the authors proposed an approach based on distributed *multi-agent reinforcement learning*. However, whenever the number of UEs change, these solutions either require a new learning procedure [6], [10] or to entirely redesign and retrain the architecture of the NNs [8]. With the complexity associated to the re-computation or re-learning procedures, such approaches are unsuitable to dynamic networks, characterized by a frequent change of the radio environment due to *e.g.* mobility or the arrival and departure of UEs.

Here, to overcome the inadequacy of state-of-the-art solutions, we first cast the user association problem as a multi-agent reinforcement learning problem, aimed at optimizing predefined network utility functions common to all UEs. Then, by conveniently adapting to this context a *neural attention mechanism*, we successfully design a global policy network architecture (PNA) that is transferable among UEs. Given this PNA, UEs learn a common association policy leveraging their local (and global, if available) observations. The learned policy has *zero-shot generalization capability*, thus considerably reducing the computational complexity of the user association task. Indeed, thanks to the proposed architecture, a policy learned in a specific deployment can be transferred to another one without requiring substantial additional training procedure. Consequently, as desired, the proposed mechanism adapts well and by design to variations in the number of UEs or changes in the geometry of the network (namely the geographical positions of the UEs). Moreover, the proposed mechanism

This work has been partially supported by the CPS4EU Project, Nr. 826276 and the H2020 RISE-6G Project.

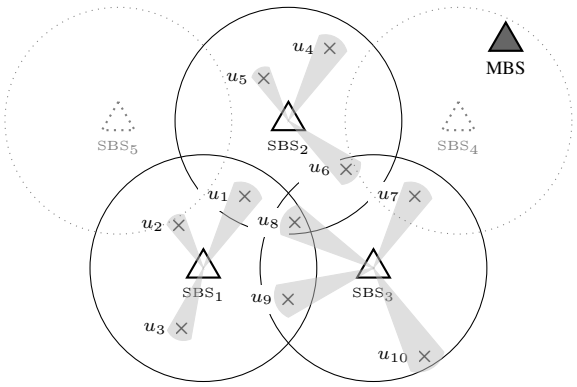

Fig. 1. Network topology for $N_s = 3$ SBSs, 1 MBS, and $K = 10$ UEs.

incorporates channels' and UEs' traffic dynamics during the training phase to foster better adaptability to the variations of radio channel quality and requested quality of service (QoS). Finally, our proposed solution can be implemented either in a centralized or in a distributed manner to trade-off computational complexity and/or signaling overhead.

II. System Model and Problem Formulation

A. Network Model

We consider the system model of Fig. 1 as in [10]: we focus on downlink communications in a network of $K(t)$ UEs located at time t in a region of the bi-dimensional Euclidean space, covered by N_s mmWave small base stations (SBSs) and a sub-6 GHz macro base station (MBS), to enable ubiquitous network coverage. Let $\mathcal{A} = \{0, 1, \ldots, N_s\}$ be the set of BSs, where 0 indexes the MBS, and $\mathcal{U}(t) = \{1, 2, \ldots, K(t)\}$ be the set of UEs in the network. We call a network deployment $\mathcal{D}(t)$, a collection of positions of all UEs in the network:

$$\mathcal{D}(t) = \{(x_j(t), y_j(t)), j \in \mathcal{U}(t)\}, \tag{1}$$

where $x_j(t)$ and $y_j(t)$ denote respectively the two coordinates of UE j in deployment $\mathcal{D}(t)$, expressed with respect to a reference system common to all UEs and BSs.

We denote with $\mathcal{A}_j(t) = \{i, d_{i,j}(t) \leq R_0\} \cup \{0\} \subseteq \mathcal{A}$ the subset of BSs that can be associated with UE j at time t. Here, R_0 is the SBSs' coverage range and $d_{i,j}(t)$ is the distance from UE j to SBS i, which we assume to be able to support at most N_i UEs simultaneously. Also, we assume that a UE is associated to exactly one BS at a time and each BS i communicates to its served UEs using equal transmit power. We adopt the Friis propagation loss model [11], according to which the power received by a UE, P^{Rx}, is given as a function of the distance d between the UE and its serving BS:

$$P^{\mathrm{Rx}}(d) = h P_s^{\mathrm{Tx}} G_s^{\mathrm{Tx}} G_s^{\mathrm{Rx}} C_s d^{-\eta_s}, \quad s \in \{\mathrm{MBS}, \mathrm{SBS}\}. \tag{2}$$

Here, C_s denotes the path-loss constant, η_s is the path-loss exponent, P_s^{Tx} is the transmit power *w.r.t.* BS s and h denotes the fading coefficient. We assume m-Nakagami fading for UE-SBS links whereas UE-MBS links experience Rayleigh fading, which is a special case of m-Nakagami, where $m = 1$. The gains of the transmitter and receiver antennas *w.r.t.* BS s are G_s^{Tx} and G_s^{Rx} respectively. In addition, we assume that mmWave SBSs allocate all the available band to their served UEs, whereas the MBS equally shares its band across its UEs. Also, we assume that UEs and BSs perform beam steering and training in advance and ignore their impact when optimizing user association. Finally, we assume that there exists a central controller, collocated with the MBS, able to collect and forward information to the UEs.

B. Problem formulation

At time t, each UE j requests a data rate $D_j(t)$ from its serving BS i to satisfy a certain QoS. It then experiences a signal-to-interference plus noise ratio $\mathrm{SINR}_{i,j}$, which comprises both intra-cell and inter-cell interference. We say that UE j's QoS is fully satisfied at time t, if the achievable data rate $R_{i,j}(t) = B_{i,j} \log_2(1 + \mathrm{SINR}_{i,j}(t))$, given by the Shannon capacity of the channel between UE j and BS i, is greater than $D_j(t)$. Therefore, the effective communication rate between UE j and its serving BS equals $\min(R_{i,j}(t), D_j(t))$. Hence, we define an α-fair *network utility function* [12] as follows:

$$\begin{aligned} R(t) &= \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{U}(t)} x_{i,j}(t) U_\alpha(\min(R_{i,j}(t), D_j(t))), \\ &= \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{U}(t)} x_{i,j}(t) U_\alpha(\kappa_{i,j}(t) D_j(t)), \end{aligned} \tag{3}$$

where $x_{i,j}(t) = 1$ indicates that UE j is associated with BS i at time t; otherwise $x_{i,j}(t) = 0$ and $\kappa_{i,j}(t) = \min\left(1, \frac{R_{i,j}(t)}{D_j(t)}\right) \in [0, 1]$ indicates the QoS satisfaction of UE j *w.r.t.* its associated BS i, which is fully satisfied when $\kappa_{i,j}(t) = 1$. $U_\alpha(\cdot)$ is the α-fair utility function given in [12, Section 2.2] as follows:

$$U_\alpha(x) = \begin{cases} (1-\alpha)^{-1} x^{1-\alpha}, & \text{if } \alpha \geq 0 \text{ and } \alpha \neq 1, \\ \log(1+x), & \alpha = 1. \end{cases} \tag{4}$$

Given a network deployment $\mathcal{D}(t)$, we formulate the user association problem to maximize (3) at time t as follows:

$$\underset{\{x_{i,j}(t)\}}{\text{maximize}} \; R(t), \tag{5a}$$

$$\text{subject to } x_{i,j}(t) \in \{0, 1\}, \; \forall i, j, \tag{5b}$$

$$\sum_{j \in \mathcal{U}(t)} x_{i,j}(t) \leq N_i, \; \forall i \in \mathcal{A} \backslash \{0\}, \tag{5c}$$

$$\sum_{i \in \mathcal{A}_j(t)} x_{i,j}(t) = 1, \; \forall j \in \mathcal{U}(t). \tag{5d}$$

Constraint (5b) indicates that the $x_{i,j}(t)$ are binary variables. The number of resources available at each SBS is limited; this is highlighted in (5c), by constraining the number of UEs simultaneously associated with a given SBS i, to be lower than N_i. Finally, (5d) ensures that, in our setting, each UE is associated with exactly one BS. Depending on α, problem (5) guarantees different fairness criteria in the user association. In particular, we will focus on $\alpha = 0$ and $\alpha = 1$, corresponding respectively to sum-rate maximization and proportional fairness, widely used in the literature [13].

Although (5) appears as a standard user association problem, solutions of the literature are often scenario-specific or deployment-specific. In other words, they assume either a pre-sized or a fixed set of static UEs. Here, we are interested in a

different approach: first, we are looking for a policy that can be applied at each time t by a user, based on its instantaneous observations of the environment. Then, taking into account the targeted optimization objective, we are interested in an association policy, which, once learned, is also transferable and capable of solving problem (5) at each time t regardless of the location and the number of UEs in the network, without the need of being relearned. This policy must be able to adapt to the departure or arrival of UEs from and in the network, as both events have an impact on the optimal user association. Also, a policy learned in a scenario of K_1 UEs has to be effectively applicable to a scenario of $K_2 \neq K_1$ UEs without additional training. For this purpose, the architecture of the association policy needs to be transferable, as well as the learned policy.

III. Proposed Transferable User Association policy

A. On transferable policy network architecture

In order for the policy architecture to be transferable, an adequate design of the PNA components is required. Our objective, in fact, is to construct a policy architecture whose size does not vary with the number of UEs in the network, which is bound to change over time. In the following, we will describe the policy network architecture of Fig. 2, which allows the transferability of the association policy.

1) UE local observation encoding: In this study, we assume that at each time t, each UE j can estimate the received signal strength (RSS) and the corresponding angle of arrival (AoA) *w.r.t.* its surrounding BSs. We denote with $\mathrm{RSS}_{i,j}$ and $\vartheta_{i,j}$ the estimated RSS and AoA of UE j *w.r.t.* BS i, respectively. Moreover, as in [10], a UE receives an acknowledgment (ACK) signal whenever its connection request succeeds ($\mathrm{ACK}_j = 1$) or is denied ($\mathrm{ACK}_j = 0$), which may happen due to the limited resources available at each BS (5c); we call this event a *collision*. When it happens, each BS selects among the colliding UEs the best ones to associate with, according to their association probability that we define later. Next, we define UE j local state, $\mathbf{o}_j^{\mathrm{L}}(t)$, as follows:

$$\mathbf{o}_j^{\mathrm{L}}(t) = \Big\{ a_j(t-1), R_{a_j(t-1),j}, R(t-1), \mathrm{ACK}_j, \{\mathrm{RSS}_{i,j}(t)\}_{i\in\mathcal{A}_j(t)}, \{\vartheta_{i,j}\}_{i\in\mathcal{A}_j(t)} \Big\}. \tag{6}$$

Here, $R_{a_j(t-1),j}$ represents the achievable rate when UE j is associated with the BS indexed by $a_j(t-1)$[1]. Note that the size of $\mathbf{o}_j^{\mathrm{L}}(t)$ does not depend on the number of UEs, in sharp contrast with [8]. Then, we obtain the n-dimensional local encoding vector $\mathbf{u}_j(t) = f(\mathbf{o}_j^{\mathrm{L}}(t))$, where $f : \mathbb{R}^l \to \mathbb{R}^n$ is a NN, and l is the size of the vector obtained after the concatenation of the elements in $\mathbf{o}_j^{\mathrm{L}}(t)$.

2) UE global observation encoding: After taking an action $a_j(t)$, the controller can encode for UE j some meaningful global state (i.e. macro) information $\mathbf{o}_j^{\mathrm{G}}(t)$ such as the estimated position of UEs of interfering links, i.e., of active mmWave links, the load of each BS, etc. However, embedding more information does not necessarily imply performance improvement. Indeed, as the agent's state space also increases, more exploration is required to discover the intrinsic state/action relation at the risk of misleading the agent. In our scenario, we consider that the information about the actual rate perceived by each UE j and the position of the potential interferers, i.e., the set of UEs $\mathcal{N}_j(t)$, susceptible to impact the association decision of UE j through the interference resulting from their communications. Here, we consider $\mathcal{N}_j(t)$ as the k-nearest neighbors of UE j, whose size may vary with time, however solutions based on local interaction graphs can be considered [4]. Hence, we define $\mathbf{o}_j^{\mathrm{G}}(t)$ as:

$$\mathbf{o}_j^{\mathrm{G}}(t) = \Big\{ \varsigma_l = \big[x_l(t), y_l(t), R_{a_l(t-1),l}\big], \quad l \in \mathcal{N}_j(t) \Big\}, \tag{7}$$

from which, we construct UE j global state encoding $\mathbf{v}_j(t)$.

Fixed-size encoding. A naive solution to construct $\mathbf{v}_j(t)$ is to first concatenate all elements in $\mathbf{o}_j^{\mathrm{G}}(t)$ resulting in a vector of size $m(t) = 3 \times \mathrm{card}(\mathcal{N}_j(t))$. Then, we obtain the local encoding vector $\mathbf{v}_j(t) = g(\mathbf{o}_j^{\mathrm{G}}(t))$, where $g : \mathbb{R}^m \to \mathbb{R}^n$ is also a NN. However, such an approach i) has limited scalability as the size of $\mathbf{o}_j^{\mathrm{G}}(t)$, i.e., $m(t)$ varies with the number of UEs, especially in the neighborhood, and ii) requires ordering elements prior to concatenation, preventing from transferability.

Order-agnostic and size-variable encoding. An efficient solution to the problem should be agnostic of the ordering in $\mathbf{o}_j^{\mathrm{G}}(t)$. Moreover, in order to build a scalable and transferable architecture, the size of $\mathbf{v}_j$ should be independent of the length

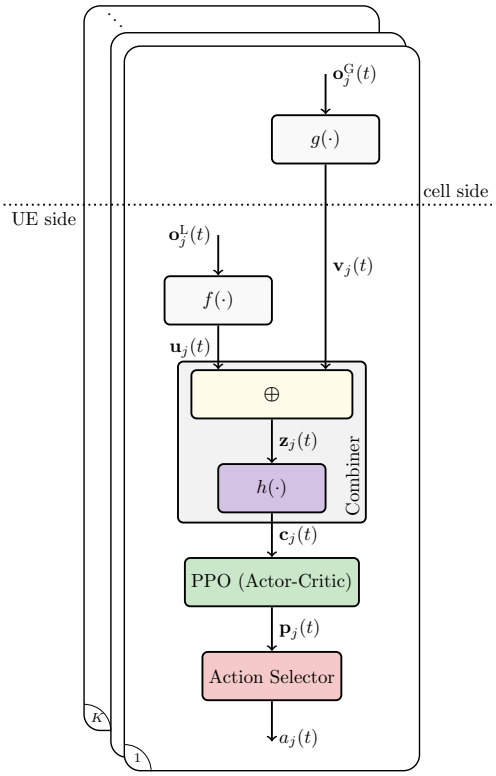

Fig. 2. UE association policy network architecture, shared across all UEs.

[1] $R(t-1)$ is local in the sense that it is related to the previous time step, already available at the UE side.

of $\mathbf{o}_j^{\mathrm{G}}(t)$, thus, the size of UE j neighborhood. To satisfy these properties, we adopt ideas from the *dot-product attention mechanisms* in [14]. Hence, let $\mathbf{k}_j = g_k(\varsigma_j)$, $\mathbf{q}_j = g_q(\varsigma_j)$, and $\boldsymbol{\nu}_j = g_\nu(\varsigma_j)$, where $g_k,\ g_q,\ g_\nu : \mathbb{R}^3 \to \mathbb{R}^n$ are also encoding functions (e.g., neural networks), and $\mathbf{k}_j, \mathbf{q}_j, \boldsymbol{\nu}_j$ denote the *key*, the *query* and the *value* associated with UE j, respectively. For a given UE j, we compute for each UE in its neighborhood $\mathcal{N}_j(t)$ a weight (or score) $\alpha_{k,j}$

$$\alpha_{k,j} = \mathbf{softmax}\left(\left[\frac{\mathbf{q}_k \mathbf{k}_j^T}{\sqrt{n}}\right]_{k\in\mathcal{N}_j(t)}\right). \tag{8}$$

Here, $\mathbf{softmax}(\cdot)$ is the softmax function also known as the normalized exponential function. Let $\boldsymbol{\alpha}_j = [\alpha_{k,j}, k \in \mathcal{N}_j]$. The vector $\boldsymbol{\alpha}_j$ represents the interaction of UE j with its neighbors. Then, we compute the encoding $\mathbf{v}_j$ by aggregating all values' information from the neighborhood as follows:

$$\mathbf{v}_j = \sum_{k\in\mathcal{N}_j(t)} \alpha_{k,j}\boldsymbol{\nu}_k. \tag{9}$$

By construction, the size of $\mathbf{v}_j$ in (9) is invariable with the size of $\mathcal{N}_j(t)$. Only its value can change depending of the aggregated information. That is to say, whenever the number of UEs varies, there is no need to change the PNA.

3) Local and global information combining: Now, once we obtain the UE local and global encoding vector, they are merged together to build its context understanding vector $\mathbf{c}_j(t) = h(\mathbf{z}_j(t))$, i.e., its perception of the radio environment, where $\mathbf{z}_j(t) = \mathbf{u}_j(t) \oplus \mathbf{v}_j(t)$, with $\oplus$ denoting concatenation operation and $h : \mathbb{R}^{2n} \to \mathbb{R}^n$ is also taken here, as a NN.

Now, given the context vector $\mathbf{c}_j(t)$, the goal of the learning agent j at each instant t, is to define an *association probability vector* $\mathbf{p}_j(t) = [p_{0,j}, \ldots, p_{N_s,j}] \in [0,1]^{N_s+1}$ with $\sum_{i\in\mathcal{A}} p_{i,j} = 1$ and $p_{i,j} = 0$ for all $i \notin \mathcal{A}_j(t)$. Then, the UE's action $a_j(t)$, which corresponds to a connection request towards the BS indexed by $a_j(t)$ in $\mathcal{A}_j(t)$, is sampled from the distribution characterized by the $p_{i,j}$. Thus, the learning problem consists in deriving an association policy that optimizes $\mathbf{p}_j(t)$, such that sampling from it maximizes (3).

Note that in this architecture, UEs' agents share the same model, i.e., $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are common to all UEs. This setting does not preclude UEs from taking different actions as they do not observe the same inputs. In contrast, sharing the parameters among UEs enables a better skill transfer since there is only a unique policy (in contrast to having one policy per UE as in [8]), which can be efficiently and simultaneously trained with the experiences of all UEs in a MARL framework using *proximal policy optimization* (PPO) [15].

B. Proximal policy optimization

In a MARL system, agents learn by interacting with a shared environment by making decisions following a Markov Decision Process (MDP). In MDP, the action $a_j(t)$ of an agent j in a given state $\mathbf{s}_j(t)$ leads it to the next state $\mathbf{s}_j(t+1)$ and results in a reward $r_j(t)$. From the underlying *experience* $e_j(t) = \{\mathbf{s}_j(t), a_j(t), r_j(t), \mathbf{s}_j(t+1)\}$, the agent learns its policy $\pi_{j,\mathbf{w}}(\cdot|\cdot)$, parameterized by $\mathbf{w}$, the set of PNA parameters, where $\pi_{j,\mathbf{w}}(a_j|\mathbf{s}_j) = p_{a_j(t),j}$ is the probability that agent j takes action (or requests connection) a_j in state $\mathbf{s}_j$, to maximize an accumulated long-term γ-discounted reward $G_j(t) = \sum_{\tau=t+1}^{T_e} \gamma^{\tau-t-1} r_j(\tau)$ over an *episode* - a new network deployment - of duration T_e: $\pi^*_{j,\mathbf{w}} = \underset{\pi_j}{\operatorname{argmax}}\, \mathbb{E}_t\left[G_j(t)\right]$. In our study, we consider the particular case of *cooperative* MARL [16], i.e., UEs share the same reward, hence, they are assigned to the same objective of maximizing the network utility function: $r_j(t) = R(t),\ \forall j$. Moreover, UEs also share the same policy parameters, i.e., $\pi_{j,\mathbf{w}} = \pi_{\mathbf{w}},\ \forall j$.

In general MARL, an agent has only access to a partial observation $\mathbf{o}_j(t) = \left\{\mathbf{o}_j^{\mathrm{L}}(t), \mathbf{o}_j^{\mathrm{G}}(t)\right\}$ of the actual state $\mathbf{s}_j(t)$, which is unknown, resulting in partially observable MDP [17]. Moreover, MARL is subject to non-stationarities due to simultaneous interactions of agents with the environment, which make the learning process more complex. In the literature, *policy gradient* algorithms are used to solve this problem [18]. We use an actor-critic mechanism to iteratively update the policy parameters $\mathbf{w}$ to minimize the ϵ-clipped proximal loss:

$$\mathcal{L}(\mathbf{w}) = \mathbb{E}_\pi\left[\min\left(\zeta(\mathbf{w})\hat{A}, \mathbf{clip}\left(\zeta(\mathbf{w}), 1-\epsilon_1, 1+\epsilon_2\right)\hat{A}\right)\right], \tag{10}$$

where $\mathbf{clip}(x,a,b) = \min(\max(x,a),b)$, $\hat{A}(a_j, \mathbf{o}_j)$ denotes the advantage estimator, which measures the advantage of selecting a given action in a given state, that we estimate using one step Temporal Difference error [18]. $\zeta(\mathbf{w}) = \frac{\pi_{\mathbf{w}}(a_j|\mathbf{o}_j)}{\pi_{\mathbf{w}_{\mathrm{old}}}(a_j|\mathbf{o}_j)}$ is the probability ratio between current and previous update. By introducing the clipping effect, PPO pessimistically ignores updates (possibly destructive) that will lead to high changes between policy updates.

Hysteretic PPO. Note that in vanilla PPO, $\epsilon_1 = \epsilon_2$. However, in multi-agent environments, an agent should not be pessimistic in the same way for both *positive* ($\zeta(\mathbf{w}) > 1$) and *negative* ($\zeta(\mathbf{w}) < 1$) updates. In fact, due to the interaction of multiple agents with the environment and the common reward of the cooperative framework, an agent may receive a lower reward because of the bad behavior of its teammates. This may cause the user to change its policy at the risk to misleading it. To overcome this issue, following the concept of hysteretic Q-learning in [19], we introduce *hysteretic proximal policy optimization*, where we use ϵ_1 and ϵ_2 for negative and positive updates, with $\epsilon_1 < \epsilon_2$. In this way, an agent gives more importance to updates that improve its policy rather than to ones that worsen it. This setting is particularly important when agents do not have equal contribution to the team's reward and for decentralized learning.

Note that the association policy can be efficiently trained in a centralized way with the experience of all agents or in a decentralized way, by leveraging approaches presented in [20].

To further make learning robust against the variability of the number of UEs over time, we introduce a *UE dropout mechanism* with rate p_0, corresponding to the Bernoulli probability of a UE to be masked out in a given training episode, thus, appearing as non-existent in the cell for the others UEs.

On complexity. In contrast to previous work where each UE learns its own specific policy without transferability [10], here we have only one global policy that can be transferred to

TABLE I
SIMULATIONS PARAMETERS [10]

	Macro cell	Small cell
Parameters	Values	
Carrier frequency f_s	2.0 GHz	28 GHz
Path loss constant C_s	$(c/4\pi f_s)^2$, $c = 3 \times 10^8 \text{ms}^{-1}$	
Bandwidth	10 MHz	200 MHz
Thermal noise, N_0	-174 dBm/Hz	
Noise figure	5 dB	0 dB
Shadowing variance	9 dB	12 dB
TX power, P^{Tx}	46 dBm	20 dBm
Antenna gain, $G^{\text{Tx}}_{\max}$ / $G^{\text{Rx}}_{\max}$	17 dBi / 0 dBi	[10, diag. 2]
Radius, R_0		50 m
Back-lobe gain		-20 dB
Path-loss exponent, η_s	3.76	2.5
Inter-cell distance		$1.2 \times R_0$
AoA error $\sim \mathcal{N}(0, \sigma^2_{\text{AoA}})$	2°	

any UE in the network even new ones, thus considerably reducing the computation complexity. Also by using attention mechanism instead of Long Short Term Memories (LSTMs), we considerably reduce the PNA architecture. The counterpart is the aggregation of information in (9). However, this process can be viewed as a message passing between UEs, where they only need to exchange their queries and values with BSs and only when there is a considerable change in the network (*e.g.* arrival of new UEs) to limit signaling.

IV. NUMERICAL RESULTS

In our simulations, we consider $K_0 = 15$ UEs randomly located in a bi-dimensional region, under the coverage of $N_s = 3$ SBSs working at mmWave frequencies with a carrier frequency of 28 GHz, and one MBS communicating at 2 GHz, however, our solution can be leveraged for applications using different technologies such as WiFi or LiFi.. Also, we consider three types of service corresponding to an average data rate demand $\overline{D}_j \in \{5, 200, 1500\}$ Mbps. We assume that the traffic request of a UE j is a random variable, which follows a Poisson distribution with intensity $\overline{D}_j = \mathbb{E}[D_j(t)]$. Additional simulation parameters can be founded in Table I.

Learning parameters. Since all UEs share the same policy network, $\mathcal{A}$ coincides with the action space. However, a UE j can only be associated with BSs in $\mathcal{A}_j(t) \subseteq \mathcal{A}$. Accordingly, unauthorized actions or connection requests $a_j(t) \notin \mathcal{A}_j(t)$ are redirected towards the MBS, i.e., they appear as connection requests to the MBS. We fixed the size of the encoding functions $n = 128$. All encoding functions are composed of only one hidden multi-layer perceptron (MLP) of n neurons. Both actor and critic comprise also one MLP with $2n$ neurons. All layers use a rectifier linear unit activation. We set the learning rate μ to 10^{-4} and the discounting factor $\gamma = 0.6$. Unless specified, we empirically fix the clipping factors to $\epsilon_1 = 0.01$, $\epsilon_2 = 0.5$, the time horizon to $T_e = 250$ and the UE dropout probability to $p_0 = 0.95$. Also, we limit UE j's neighborhood $\mathcal{N}_j$ to its k-nearest neighbors, where $k \leq 15$.

Benchmarks. As a comparison, we consider the same benchmarks as in [10], i.e., the Max-SNR algorithm, which associates UEs on the basis of links with the maximum SNR, and the centralized heuristic algorithm, which consists in associating UEs, starting from the links with the maximum SNR, and in an iterative way as long as it increases the network utility. Originally proposed in [6], the centralized heuristic algorithm is shown to exhibit good performance, specifically in interference-limited network. Therefore, we use it as a baseline solution in lieu of the optimal solution, infeasible here, due to the network size. However, we recompute the heuristic algorithm every time the network changes, which is also cumbersome. To assess the convergence performance of the proposed algorithm, we define $r_d(t) = \overline{R}^{\text{Trans. RL}}(t) - \overline{R}^{\text{Heur.}}(t)$, which corresponds to the difference of the average reward over an episode reached by the proposed algorithm compared to the centralized heuristic approach. Also, we represent on the histograms, the average performance over 500 random deployments of UEs.

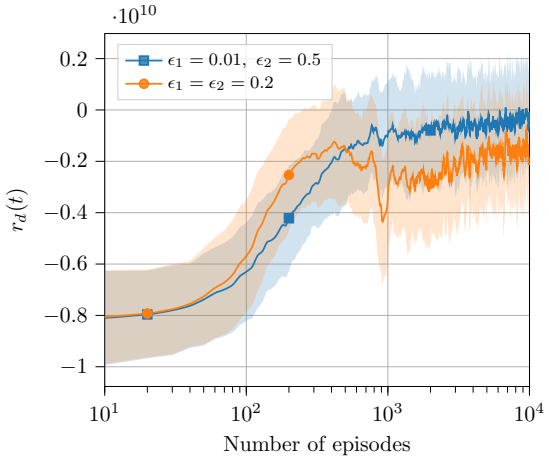

Fig. 3. Effect of the hysteretic clipping factors on the convergence. $\alpha = 0$ and $D_j(t) = \infty$, $\forall j$. Averaged on a 100-sized rolling window.

A. Impact of the hysteretic clipping factors on convergence

Here, we evaluate the impact of clipping factors ϵ_1 and ϵ_2 on the convergence. Fig. 3 shows the evolution of $r_d(t)$ in two settings: $\epsilon_1 = \epsilon_2 = 0.2$, corresponding to the setting of the vanilla PPO proposed in [15], and our empirically optimized hysteretic setting $\epsilon_1 = 0.01$, $\epsilon_2 = 0.5$. We show that by simply introducing a hysteretic effect in the clipping factors, we notably improve the stability and the learning performance, reaching the same performance as the heuristic algorithm (as $r_d(t)$ converges on average to zero).

B. Policy Transferability Property: Zero shot generalization

To assess how transferable the proposed algorithm is, we train the PNA for a reference number of users, $K_0 = 15$ and $N_i = 3$, $\forall i$. Then we evaluate on Fig. 4 and 5, the performance of the trained model for different network deployments with a variable number of UEs $K \in \{10, 20, 25, 30\}$, including changes in the UEs' position and traffic dynamic. Fig. 4 shows that when we consider the network traffic, the proposed transferable solution clearly outperforms the two benchmarks. Even when the number of UEs doubles from $K_0 = 15$ to $K = 30$, our solution yields 102.1%, 66.66% network sum-rate increase *w.r.t.* the max-SNR and heuristic algorithms, respectively. Moreover, an additional feature of the proposed

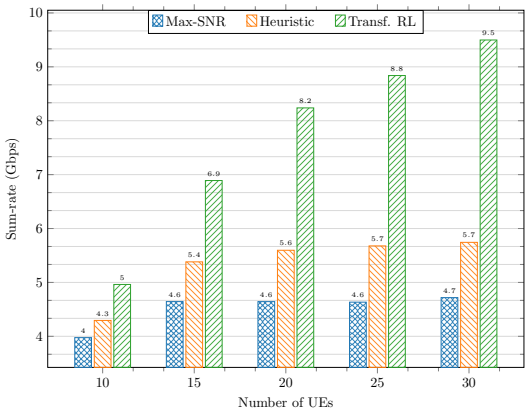

Fig. 4. Transferability for case $\alpha = 0$ and with network traffic. The PNA is initially trained for 15 UEs, and the performance evaluated for different K.

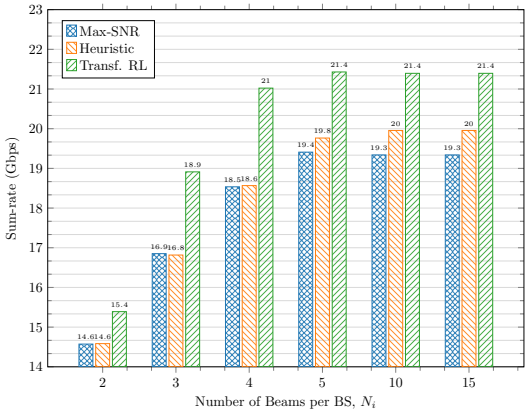

Fig. 5. Generalizability for case $\alpha = 1$ (no traffic). Here we fix $K = 15$ and vary the number of beams N_i.

architecture, is that even when the number of beams available per BS later changes (*w.r.t.* initial training point, fixed to $N_i = 3$), which impacts the collision events, the algorithm still adapts to maintain the system's performance. Indeed, in Fig. 5 where we evaluate the performance of the algorithms for different N_i, we can observe that as N_i increases, implying less and less collisions since K is fixed, the algorithm keeps outperforming the two benchmarks. When N_i becomes greater than 5, i.e., $\sum_{i=1}^{3} N_i > K = 15$, there is no improvement in the sum-rate as there are enough beams to serve all UEs.

V. Conclusions and Perspectives

In this work, we investigated the problem of transferability of user association policies for 5G and beyond networks. We come out with a novel policy architecture and a learning mechanism that enable users to cooperatively learn a robust and transferable user association policy. Our proposed solution exploits neural attention and deep multi-agent reinforcement learning mechanisms, where agents leverage local and, if available, global observations to optimize network utility functions. We achieve transferability. Indeed, with our solution, a policy learned in a given scenario can be transferred with zero-shot generalization capability, i.e. without any additional training. Our numerical results showed that the proposed transferable solution provides large gains, indeed doubling the network sum-rate compared to state-of-the-art approaches. The observed benefit is due to the transferability feature, and by jointly considering network traffic and radio channels dynamic during optimization. In future work, results of this study will be exploited for applications in Multi-access Edge Computing and its possible extension will be explored for semantic and goal oriented communication [21].

References

[1] E. Calvanese Strinati *et al.*, "6G: The Next Frontier: From Holographic Messaging to Artificial Intelligence Using Subterahertz and Visible Light Communication," *IEEE Veh. Tech. Mag.*, vol. 14, pp. 42–50, Sep. 2019.

[2] A. De Domenico *et al.*, "Making 5G Millimeter-Wave Communications a Reality [Industry Perspectives]," *IEEE Wireless Communications*, vol. 24, pp. 4–9, Aug 2017.

[3] 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 Transactions on Networking*, vol. 23, pp. 836–850, June 2015.

[4] Y. Liu, X. Fang, M. Xiao, and S. Mumtaz, "Decentralized Beam Pair Selection in Multi-Beam Millimeter-Wave Networks," *IEEE Transactions on Communications*, vol. 66, pp. 2722–2737, June 2018.

[5] A. Alizadeh and M. Vu, "Early acceptance matching game for user association in 5g cellular hetnets," in *IEEE GLOBECOM*, pp. 1–6, 2019.

[6] P. Zhou *et al.*, "Deep Learning-Based Beam Management and Interference Coordination in Dense mmWave Networks," *IEEE Transactions on Vehicular Technology*, vol. 68, pp. 592–603, Jan 2019.

[7] R. Liu, M. Lee, G. Yu, and G. Y. Li, "User association for millimeter-wave networks: A machine learning approach," *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 4162–4174, 2020.

[8] N. Zhao, Y. Liang, D. Niyato, Y. Pei, and Y. Jiang, "Deep Reinforcement Learning for User Association and Resource Allocation in Heterogeneous Networks," in *Proc. IEEE GLOBECOM*, pp. 1–6, Dec 2018.

[9] M. Sana, A. De Domenico, and E. Calvanese Strinati, "Multi-Agent Deep Reinforcement Learning based User Association for Dense mmWave Networks," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, HI, USA, Dec 2019.

[10] M. Sana, A. De Domenico, W. Yu, Y. Lostanlen, and E. Calvanese Strinati, "Multi-Agent Reinforcement Learning for Adaptive User Association in Dynamic mmWave Networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6520–6534, 2020.

[11] T. Bai and R. W. Heath, "Coverage and Rate Analysis for Millimeter-Wave Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 1100–1114, 2015.

[12] R. Srikant and L. Ying, *Communication Networks: An Optimization, Control and Stochastic Networks Perspective*. USA: Cambridge University Press, 2014.

[13] D. Liu *et al.*, "User Association in 5G Networks: A Survey and an Outlook," *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1018–1044, 2016.

[14] A. Vaswani *et al.*, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms.," *in Proc. CoRR*, 2017.

[16] L. Buşoniu, R. Babuška, and B. De Schutter, "Multi-agent Reinforcement Learning: An Overview," in *Innovations in multi-agent systems and applications-1*, pp. 183–221, Springer, 2010.

[17] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian, "Deep Decentralized Multi-task Multi-Agent Reinforcement Learning under Partial Observability," in *Proc. International Conference on Machine Learning (ICML)*, vol. 70, pp. 2681–2690, PMLR, 06–11 Aug 2017.

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

[19] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, "Hysteretic Q-Learning: An Algorithm for Decentralized Reinforcement Learning in Cooperative Multi-agent Teams," in *Proc. International Conference on Intelligent Robots and Systems (IEEE/RSJ)*, pp. 64–69, 2007.

[20] E. Wijmans *et al.*, "DD-PPO: Learning Near-Perfect PointGoal Navigators from 2.5 Billion Frames," in *Proc. ICLR*, 2020.

[21] E. Calvanese Strinati and S. Barbarossa, "6G networks: Beyond Shannon towards semantic and goal-oriented communications," *Computer Networks*, vol. 190, p. 107930, 2021.