2022 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-6654-4266-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/WCNC51071.2022.9771660

Deep Reinforcement Learning for Dynamic Clustering and Resource Allocation in Smart-Duplex Networks

Dan Wang and Chuan Huang

***Abstract*—This paper considers an ultra dense network (UDN) with smart-duplex (SD), which allows the base stations (BSs) to flexibly switch between half-duplex (HD) and full-duplex (FD) modes over time. All the small cells are divided into several clusters, where the BSs in the same cluster jointly serve their users. A Markov decision process (MDP) problem is formulated to maximize the average weighted sum of network throughput and clustering cost for all clusters. To approximately solve this problem, we first adopt an affinity propagation method to determine the number of clusters and the center of each cluster. Then, by treating small cells as agents, the original MDP problem is proved to be equivalent to a multi-agent MDP to maximize the average reward of all small cells. Next, a multi-agent deep reinforcement learning (DRL) algorithm is proposed to jointly implement the dynamic clustering for the non-center small cells, resource allocation, and duplex mode selection. Simulation results show that SD has prominent advantages over both the HD and FD modes in UDNs, and the proposed algorithm outperforms other clustering schemes under the considered scenarios.**

I. Introduction

Ultra dense networks (UDNs), which support network density with ten time higher than before [1], have been regarded as one of key technologies to meet the increasing demands of data transmissions for future networks [2]. As the network becomes larger, how to make full use of the resource in UDNs over space, time, and frequency becomes the new challenge.

For the conventional UDNs, BSs and mobile users operate under the half-duplex (HD) mode, i.e., they transmit and receive signals at different time slots or frequency bands [3]. The authors in [4] investigated a weighted sum-rate maximization problem for a HD UDN by clustering all the BSs into multiple clusters, where only the BSs in same cluster need to coordinate to jointly serve the nearby users. On the other hand, full-duplex (FD), which allows the transmitters and receivers to simultaneously operate in the same frequency band [3], has been regarded as a potential technique to improve the spectrum efficiency of UDNs. The authors in [5] proposed a joint user access and resource allocation problem in a FD UDN, where K-means clustering is adopted to reduce the computational complexity by decoupling the original problem into the intra-cluster user access and subchannel allocation problem and the inter-cluster power control problem. However, the interference distribution of the FD networks is more complicated than that of the HD networks under the same densities of BSs and mobile users, e.g., the self-interference (SI) and the inter-cell and intra-cell interference from the uplink and downlink transmissions in the FD networks are much more severe than those in the HD networks [3]. Therefore, it is interesting to study the flexible switching scheme between FD and HD at the BSs, i.e., smart-duplex (SD), in the wireless networks according to the dynamic environment.

The aforementioned methods only focus on static networks where the CSI is constant over time, and are not scalable to dynamic wireless environment. In the recent years, deep reinforcement learning (DRL) has been well studied to solve the large network design problems with changing CSI [6], [7]. The BSs or users are treated as multiple agents to directly interact with the dynamic environment based on their local observation (e.g., the CSI, the topology of users access BSs, etc.) instead of the complete information about the whole networks. To deal with the huge action and state spaces in large networks, multi-agent actor-critic (MAAC) framework, which supports the agents cooperate with each other to share their observed information during the centralized training and work independently without any interactions during the execution, was investigated to implement the distributed spectrum allocation [6]. Multi-agent deep deterministic policy gradient (MADDPG) as an extension of MAAC on the continuous action space is also adopted to solve multi-dimensional resource allocation problem for unmanned aerial vehicles assisted vehicular networks [7].

This paper proposes a dynamic clustering and resource allocation scheme for a SD-powered UDNs to maximize the average weighted sum of the network throughput and clustering cost among different clusters. With the affinity propagation (AP) method, the number of clusters and each center is firstly determined, and then the original problem is proved to be equivalent to a multi-agent MDP to maximize the average reward of all small cells by treating each small cell as one agent. A multi-agent DRL is proposed to jointly implement

D. Wang is with the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu, China, 611731. Email: wdan7@outlook.com.

C. Huang is currently with the School of Science and Engineering and Future Network of Intelligence Institute, the Chinese University of Hong Kong, Shenzhen, China, 518172, and Peng Cheng Laboratory, Shenzhen, China, 518066. Email: huangchuan@cuhk.edu.cn.

The work was supported in part by the National Key R&D Program of China with grant No. 2018YFB1800800, by NSFC with grant No. 62022070, by Shenzhen Outstanding Talents Training Fund, by Guangdong Research Project No. 2017ZT07X152, and by the Basic Research Project No. HZQB-KCZYZ-2021067 of Hetao Shenzhen-HK S&T Cooperation Zone.

the dynamic clustering for the non-center small cells, resource allocation, and duplex mode selection. Simulation results show that SD has prominent advantages over both the HD and FD modes in UDNs, and the proposed multi-agent DRL outperforms other clustering schemes under the considered dynamic scenarios.

II. System Model

We consider a UDN consisting of N randomly deployed small cells, each of which is with one BS and one pair of uplink user (UU) and downlink user (DU). All the BSs and users are equipped with single antenna. Each BS communicates with its designed users under the SD mode. All the wireless links in the considered UDNs are allocated with the same frequency band, and thus there must be co-channel interference across different small cells.

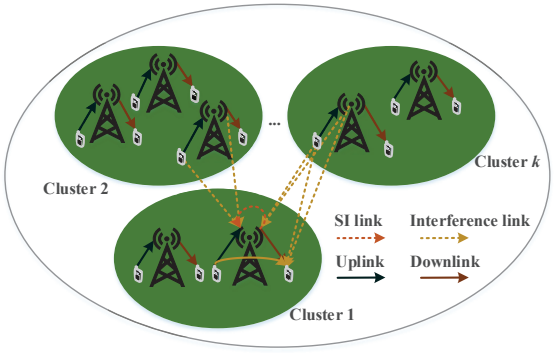

Fig. 1. Gaussian IC with two full-duplex receivers.

To avoid co-channel interference and achieve high speed transmissions, small cells are willing to form clusters. The group of the BSs in the same cluster is treated as one single virtual BS entity (VBSE) with multiple antennas (each antenna corresponds to one individual BS) to cooperatively serve all the associated users in this cluster as shown in Fig. 1. These N BSs are denoted as $B_1, \cdots, B_N$, respectively, and compose of K clusters denoted as $\mathcal{C}_1(t), \cdots, \mathcal{C}_K(t)$ at time slot t, with $1 \leq K \leq N$[1]. Let $N_k(t)$ denote the number of BSs in the k-th cluster at time slot t, and the k-th cluster is represented by $\mathcal{C}_k(t) = \{B_{k_1}, B_{k_2}, \cdots, B_{k_{N_k(t)}}\}$, with $\sum_{k=1}^{K} N_k(t) = N, t \in \mathcal{T} = \{1, \cdots, T\}$. At each time slot t, the clustering scheme is denoted as $\mathcal{C}(t) = \{\mathcal{C}_1(t), \cdots, \mathcal{C}_K(t)\}$, $\forall \mathcal{C}_k \in \mathbf{\Omega}$, with $\mathbf{\Omega}$ being the power set of $\{B_1, \cdots, B_N\}$.

[1] $K = 1$ implies that all BSs form one cluster; $K = N$ implies that all clusters are singletons. Since the number of clusters varies over time, the value of K at different time slots may be different.

A. Uplink Transmissions

At time slot t, each UU in the k-th cluster $\mathcal{C}_k(t), k \in \mathcal{K}(t) = \{1, \cdots, K\}$, sends its signal $x_{k_n}^{\mathrm{UL}}(t)$ with power $p_{k_n}^{\mathrm{UL}}(t)$ to the corresponding VBSE equipped with $N_k(t)$ receiver antennas, and the uplink transmissions in each cluster can be modeled as a multi-user single-input multi-output system [9]. The received signal $\mathbf{y}_k^{\mathrm{UL}}(t)$ at the VBSE contains the desired signals from the desired UUs in the same cluster $\mathcal{C}_k(t)$, the inter-cluster interference signals from other UUs and VBSEs in clusters $\mathcal{C}_j(t) \in \mathcal{C}(t) \backslash \mathcal{C}_k(t)$[2], the SI signals from the local transmitters, and the complex symmetric circularly Gaussian (CSCG) noise, as given in (1), where $\mathbf{w}_{k_n}^{\mathrm{DL}}(t) x_{k_n}^{\mathrm{DL}}(t)$ is the received SI at the VBSE from the local transmitters, with $x_{k_n}^{\mathrm{DL}}(t)$ and $\mathbf{w}_{k_n}^{\mathrm{DL}}(t)$ respectively being the data symbol and the precoder from the VBSE to the k_n-th DU; $x_{j_m}^{\mathrm{UL}}(t)$ and $\mathbf{w}_{j_m}^{\mathrm{DL}}(t) x_{j_m}^{\mathrm{DL}}(t)$ are the received inter-cluster interference signals at the VBSE from the j_m-th UU and the VBSE of cluster $\mathcal{C}_j$, respectively; $\mathbf{z}_k^{\mathrm{UL}}(t) \in \mathbb{C}^{N_k(t) \times 1} \sim \mathcal{CN}(0, \sigma^2 \mathbf{I}_{N_k(t)})$ is the CSCG noise vector, with $\mathbf{I}_{N_k(t)}$ being the $N_k(t)$ dimensional identity matrix; $\mathbf{h}_{k_n}^{\mathrm{UL}}(t) \in \mathbb{C}^{N_k(t) \times 1}$ is the channel coefficient from the k_n-th UU to the VBSE of cluster $\mathcal{C}_k(t)$; $\mathbf{H}_k^{\mathrm{SI}}(t) \in \mathbb{C}^{N_k(t) \times N_k(t)}$ is the SI channel coefficient for the VBSE of cluster $\mathcal{C}_k(t)$; $\mathbf{h}_{j_m,k}^{\mathrm{UU}}(t) \in \mathbb{C}^{N_k(t) \times 1}$ and $\mathbf{H}_{j,k}^{\mathrm{DU}}(t) \in \mathbb{C}^{N_k(t) \times N_j(t)}$ are the interference channel from the j_m-th UU and the VBSE of cluster $\mathcal{C}_j(t)$ to the VBSE of cluster $\mathcal{C}_k(t)$, respectively. During one time slot, the channel coefficient is treated as a constant, and across different time slots, the channel is modeled as a first-order complex Gauss-Markov process [10].

In (1), the SI signal received at the VBSE $\sum_{n \in \mathcal{C}_k(t)} \mathbf{H}_k^{\mathrm{SI}}(t) \mathbf{w}_{k_n}^{\mathrm{DL}}(t) x_{k_n}^{\mathrm{DL}}(t)$ can be partly eliminated by using some advanced SI cancellation techniques [3], and its remnant is modeled as the CSCG noise with mean zero and variance ζ^2 [11]. Then, successive interference cancellation

[2] $\mathcal{C}(t) \backslash \mathcal{C}_k(t)$ is the set of all the clusters in $\mathcal{C}(t)$ except for the k-th cluster $\mathcal{C}_k(t)$.

$$\mathbf{y}_k^{\mathrm{UL}}(t) = \sum_{n \in \mathcal{C}_k(t)} \left(\mathbf{h}_{k_n}^{\mathrm{UL}}(t) x_{k_n}^{\mathrm{UL}}(t) + \mathbf{H}_k^{\mathrm{SI}}(t) \mathbf{w}_{k_n}^{\mathrm{DL}}(t) x_{k_n}^{\mathrm{DL}}(t) \right) + \sum_{\mathcal{C}_j(t) \in \mathcal{C}(t) \backslash \mathcal{C}_k(t)} \sum_{m \in \mathcal{C}_j(t)} \left(\mathbf{h}_{j_m,k}^{\mathrm{UU}}(t) x_{j_m}^{\mathrm{UL}}(t) + \mathbf{H}_{j,k}^{\mathrm{DU}}(t) \mathbf{w}_{j_m}^{\mathrm{DL}}(t) x_{j_m}^{\mathrm{DL}}(t) \right) + \mathbf{z}_k^{\mathrm{UL}}(t). \tag{1}$$

$$R_k^{\mathrm{UL}}(t) = \sum_{n \in \mathcal{C}_k(t)} \log \left(1 + \frac{p_{k_n}^{\mathrm{UL}}(t) (\mathbf{h}_{k_n}^{\mathrm{UL}}(t))^{\mathrm{H}} \mathbf{h}_{k_n}^{\mathrm{UL}}(t)}{\sigma^2 \mathbf{I}_{N_k(t)} + \sum_{m \in \mathcal{C}_k(t), m > n} p_{k_m}^{\mathrm{UL}}(t) \mathbf{h}_{k_m}^{\mathrm{UL}}(t) (\mathbf{h}_{k_m}^{\mathrm{UL}}(t))^{\mathrm{H}} + \zeta^2 \mathbf{I}_{N_k(t)} + \mathbf{\Gamma}_k^{\mathrm{UL}}(t)} \right)$$
$$= \log \left(\det \left(\mathbf{I}_{N_k(t)} + \frac{\sum_{n \in \mathcal{C}_k(t)} p_{k_n}^{\mathrm{UL}}(t) \mathbf{h}_{k_n}^{\mathrm{UL}}(t) (\mathbf{h}_{k_n}^{\mathrm{UL}}(t))^{\mathrm{H}}}{\zeta^2 \mathbf{I}_{N_k(t)} + \sigma^2 \mathbf{I}_{N_k(t)} + \mathbf{\Gamma}_k^{\mathrm{UL}}(t)} \right) \right). \tag{2}$$

(SIC) decoder [9] is applied at the VBSE, and the decoding order is taken from 1 to $N_k(t)$. Thus, the achievable rate for uplink transmission in cluster $\mathcal{C}_k(t)$ is obtained by (2) [9], where $\mathbf{\Gamma}_k^{\mathrm{UL}}(t)$ given in (3) is the covariance matrix of the received inter-cluster interference at the VBSE from other UUs and VBSEs in clusters $\mathcal{C}_j(t) \in \mathcal{C}(t) \backslash \mathcal{C}_k(t)$.

B. Downlink Transmissions

For the downlink transmissions in cluster $\mathcal{C}_k(t)$, the VBSE precodes the data symbol $x_{k_n}^{\mathrm{DL}}(t)$ by multiplying the precoder $\mathbf{w}_{k_n}^{\mathrm{DL}}(t) = [w_{k_1 n}^{\mathrm{DL}}(t), \cdots, w_{k_{N_k} n}^{\mathrm{DL}}(t)]^{\mathrm{T}} \in \mathbb{C}^{N_k(t) \times 1}$ for each DU $k_n \in \mathcal{C}_k(t)$, and then transmits signal $\mathbf{w}_{k_n}^{\mathrm{DL}}(t) x_{k_n}^{\mathrm{DL}}(t)$ to the k_n-th DU. The downlink transmissions from the VBSE to the desired DU in the same cluster can be modeled as a multi-input single-output channel [9]. The received signal at the k_n-th DU includes the desired signals from the desired VBSE of cluster $\mathcal{C}_k(t)$, the intra-cluster interference signals from other UUs in the same cluster, the multi-user interference signals from the VBSE of cluster $\mathcal{C}_k(t)$, the inter-cluster interference signals from other UUs and VBSEs of clusters $\mathcal{C}_j(t) \in \mathcal{C}(t) \backslash \mathcal{C}_k(t)$, and the CSCG noise, as given in (4), where $\mathbf{w}_{k_n}^{\mathrm{DL}}(t) x_{k_m}^{\mathrm{DL}}(t)$ is the multi-user interference from the VBSE of cluster $\mathcal{C}_k(t)$; $x_{k_m}^{\mathrm{UL}}(t)$ is the intra-cluster interference from the k_m-th UU in the same cluster; $x_{j_m}^{\mathrm{UL}}(t)$ and $x_{j_m}^{\mathrm{DL}}(t)$ are the inter-cluster interference from the j_m-th UU and the VBSE of cluster $\mathcal{C}_j(t)$, respectively; $z_{k_n}^{\mathrm{DL}}(t) \sim \mathcal{CN}(0, \sigma^2)$ is the received CSCG noise; $\mathbf{h}_{k_n}^{\mathrm{DL}}(t) \in \mathbb{C}^{N_k(t) \times 1}$ models the propagation from the VBSE to the k_n-th DU in $\mathcal{C}_k(t)$; $h_{k_m, k_n}^{\mathrm{UD}}(t)$ and $h_{j_m, k_n}^{\mathrm{UD}}(t)$ respectively denote the interference channels from the k_m-th UU and j_m-th UU to the k_n-th DU; $\mathbf{h}_{j, k_n}^{\mathrm{DD}}(t) \in \mathbb{C}^{N_j(t) \times 1}$ is the interference channel from the VBSE of cluster $\mathcal{C}_j(t)$ to the k_n-th DU.

For the downlink transmissions, the achievable rate in cluster $\mathcal{C}_k(t)$ is computed as [9]

$$R_k^{\mathrm{DL}}(t) = \sum_{n \in \mathcal{C}_k(t)} \log \left(1 + \frac{|(\mathbf{h}_{k_n}^{\mathrm{DL}}(t))^{\mathrm{H}} \mathbf{w}_{k_n}^{\mathrm{DL}}(t)|^2}{\wp_k^{\mathrm{DL}}(t) + \sigma^2} \right), \quad (5)$$

where $\wp_k^{\mathrm{DL}}(t) = \Psi_{k_n}^{\mathrm{DL}}(t) + \Upsilon_{k_n}^{\mathrm{DL}}(t) + \Gamma_{k_n}^{\mathrm{DL}}(t)$, with $\Psi_{k_n}^{\mathrm{DL}}(t) = \sum_{m \in \mathcal{C}_k(t)} p_{k_m}^{\mathrm{UL}}(t) |h_{k_m, k_n}^{\mathrm{UD}}(t)|^2$ being the intra-cluster interference from the UUs in the same cluster, $\Upsilon_{k_n}^{\mathrm{DL}}(t) = \sum_{m \in \mathcal{C}_k(t) \backslash n} |(\mathbf{h}_{k_n}^{\mathrm{DL}}(t))^{\mathrm{H}} \mathbf{w}_{k_m}^{\mathrm{DL}}(t)|^2$ being the received multi-user interference from the VBSE of cluster $\mathcal{C}_k(t)$, and $\Gamma_{k_n}^{\mathrm{DL}}(t)$ being the inter-cluster interference $\Gamma_{k_n}^{\mathrm{DL}}(t)$ from other UUs and VBSEs of clusters $\mathcal{C}_j(t) \in \mathcal{C}(t) \backslash \mathcal{C}_k(t)$ as given in (6).

In the downlink transmissions, the transmit power of the k_n-th BS is

$$p_{k_n}^{\mathrm{DL}}(t) = \sum_{n \in \mathcal{C}_k(t)} \mathrm{Tr} \left(\mathbf{B}_{k_n}(t) \mathbf{w}_{k_n}^{\mathrm{DL}}(t) \left(\mathbf{w}_{k_n}^{\mathrm{DL}}(t) \right)^{\mathrm{H}} \right), \quad (7)$$

where $\mathbf{B}_{k_n}(t) \in \mathbb{R}^{N_k(t) \times N_k(t)} \triangleq \mathrm{Diag}\,(0, \cdots, 0, 1, 0, \cdots, 0)$, with the k_n-th diagonal element being 1. The duplex mode selection at each small cell is implemented by the power control. $p_{k_n}^{\mathrm{UL}}(t) p_{k_n}^{\mathrm{DL}}(t) = 0$ indicates the n-th small cell working on the HD mode; otherwise, it works on the FD mode, where the corresponding received SI power is denoted as $\sum_{m \in \mathcal{C}_k(t)} h_{k_m, k_n}^{\mathrm{SI}} p_{k_n}^{\mathrm{DL}}(t)$, with $h_{k_m, k_n}^{\mathrm{SI}}$ being the element in row m and column n of $\mathbf{H}_k^{\mathrm{SI}}(t)$.

C. Problem Formuation

In this subsection, the reward of each cluster at each time slot is defined as the weighted sum of the network throughput and clustering cost. Then, the average reward maximization in the long-term time scale is formulated as a MDP problem.

At the t-th time slot, the network throughput of cluster $\mathcal{C}_k(t)$ with N_k small cells is given as

$$R_{\mathcal{C}_k(t)} = R_k^{\mathrm{UL}}(t) + R_k^{\mathrm{DL}}(t), \forall k \in \mathcal{K}(t), \quad (8)$$

where $R_k^{\mathrm{UL}}(t)$ and $R_k^{\mathrm{DL}}(t)$ are given in (2) and (5).

Considering the BS cooperations in one cluster, the clustering cost consists of the signaling overhead for the CSI estimation and message decoding costs. In general, the CSI estimation overhead is proportional to the number of the links between the BSs and users, which is in the order of $O(N_k^2(t))$. In addition, the decoding cost is determined by the computational complexity of the SIC decoding, which is in the order of $O(2N_k^2(t) \log N_k(t))$ [3]. In summary, the clustering cost can be approximated by $O(N_k^2(t) \log N_k(t))$ when $N_k(t)$ is large. Then, the reward of cluster $\mathcal{C}_k(t)$ is defined as the weighted sum of network throughput and clustering cost, i.e.,

$$v_{\mathcal{C}_k(t)} = R_{\mathcal{C}_k(t)} - q_k N_k^2(t) \log N_k(t), \quad (9)$$

where $q_k \geq 0$ is a constant price for clustering.

Based on (9), our goal is to maximize the average reward for the SD UDNs in the long-term time scale by dynamically

$$\mathbf{\Gamma}_k^{\mathrm{UL}}(t) = \sum_{\mathcal{C}_j(t) \in \mathcal{C}(t) \backslash \mathcal{C}_k(t)} \sum_{m \in \mathcal{C}_j(t)} p_{j_m}^{\mathrm{UL}}(t) \mathbf{h}_{j_m, k}^{\mathrm{UU}}(t) (\mathbf{h}_{j_m, k}^{\mathrm{UU}}(t))^{\mathrm{H}} + \mathbf{h}_{j_m}^{\mathrm{DU}}(t) (\mathbf{w}_{j_m}^{\mathrm{DL}}(t))^{\mathrm{H}} \mathbf{w}_{j_m}^{\mathrm{DL}}(t) (\mathbf{h}_{j_m}^{\mathrm{DU}}(t))^{\mathrm{H}}. \quad (3)$$

$$\begin{aligned} y_{k_n}^{\mathrm{DL}}(t) = {} & (\mathbf{h}_{k_n}^{\mathrm{DL}}(t))^{\mathrm{H}} \mathbf{w}_{k_n}^{\mathrm{DL}}(t) x_{k_n}^{\mathrm{DL}}(t) + \sum_{m \in \mathcal{C}_k(t) \backslash n} (\mathbf{h}_{k_n}^{\mathrm{DL}}(t))^{\mathrm{H}} \mathbf{w}_{k_m}^{\mathrm{DL}}(t) x_{k_m}^{\mathrm{DL}}(t) + \sum_{m \in \mathcal{C}_k(t)} h_{k_m, k_n}^{\mathrm{UD}}(t) x_{k_m}^{\mathrm{UL}}(t) \\ & + \sum_{\mathcal{C}_j(t) \in \mathcal{C}(t) \backslash \mathcal{C}_k(t)} \sum_{m \in \mathcal{C}_j(t)} \left(h_{j_m, k_n}^{\mathrm{UD}}(t) x_{j_m}^{\mathrm{UL}}(t) + (\mathbf{h}_{j, k_n}^{\mathrm{DD}}(t))^{\mathrm{H}} \mathbf{w}_{j_m}^{\mathrm{DL}}(t) x_{j_m}^{\mathrm{DL}}(t) \right) + z_{k_n}^{\mathrm{DL}}(t). \end{aligned} \quad (4)$$

$$\Gamma_{k_n}^{\mathrm{DL}}(t) = \sum_{\mathcal{C}_j(t) \in \mathcal{C}(t) \backslash \mathcal{C}_k(t)} \sum_{m \in \mathcal{C}_j(t)}^{N_j} \left(|(\mathbf{h}_{j, k_n}^{\mathrm{DD}}(t))^{\mathrm{H}} \mathbf{w}_{j_m}^{\mathrm{DL}}(t)|^2 + p_{j_m}^{\mathrm{UL}}(t) |h_{j_m, k_n}^{\mathrm{UD}}(t)|^2 \right). \quad (6)$$

designing the clustering and resource allocation scheme, i.e.,

$$\max_{\{\mathcal{P}^{\mathrm{UL}},\mathcal{W}^{\mathrm{DL}},\mathcal{G}\}} \lim_{T\to\infty}\frac{1}{T}\sum_{t\in\mathcal{T}}\gamma^{t-1}\sum_{k\in\mathcal{K}(t)} v_{\mathcal{C}_k(t)} \tag{10}$$

$$\text{s.t.} \quad 0\le p_{k_n}^{\mathrm{UL}}(t)\le P_{\mathrm{UL}},\forall k,t,n, \tag{11}$$

$$0\le p_{k_n}^{\mathrm{DL}}\le P_{\mathrm{DL}},\forall k,t,n, \tag{12}$$

$$\mathcal{C}_k(t)\cap\mathcal{C}_j(t)=\emptyset,\forall k\neq j,t, \tag{13}$$

where $\mathcal{P}^{\mathrm{UL}}=\{p_{k_n}^{\mathrm{UL}}(t)\}_{k_n\in\mathcal{C}_k(t),t\in\mathcal{T}}$ denotes the set of transmit power of all UUs in T time slots; $\mathcal{W}^{\mathrm{DL}}=\{\mathbf{w}_{k_n}^{\mathrm{DL}}(t)\}_{k_n\in\mathcal{C}_k(t),t\in\mathcal{T}}$ is the set of transmission parameters from all VBSEs to DUs in the T time slots; $\mathcal{G}=\{\mathcal{C}(t)\}_{t\in\mathcal{T}}$ represents the set of clustering paradigms during T time slots; $\gamma\in[0,1]$ is a discount factor; and P_{UL} and P_{DL} are the per UU and per BS transmission power budgets for the uplink and downlink transmissions, respectively.

Remark 2.1: For problem (10)-(12), it is too costly to obtain complete state information, e.g., the cluster patterns and CSI, about the whole large network for the online resource allocation by using conventional algorithms, e.g., approximation convex optimization [4]. Hence, DRL algorithms arise, which support the devices in wireless networks to allocate resource by receiving the immediate rewards from interactions with environment. All of the DRL algorithms require the number of agents and the dimension of each design variables being constant throughout the whole time scale. However, in problem (10)-(12), each cluster is regarded as agent, and the number of the clusters as well as the dimension of design variable $\mathbf{w}_{k_n}^{\mathrm{DL}}(t)$ vary over time. Hence, it is difficult to solve problem (10)-(12) by conventional DRL algorithms.

III. Multi-Agent DRL Framework

To approximately solve problem (10)-(12) via DRL algorithm, a location-based AP algorithm is adopted to determine the number of clusters and the center of each cluster. Then, by treating each small cell as one agent, problem (10)-(12) is reformulated as a multi-agent MDP to maximize the average reward of all small cells.

A. Cluster Center Selection

Referencing by [12], a location-based AP method is employed to determine the cluster centers based on three elements, i.e., similarity S, responsibility R, and availability A.

(I) Definitions of similarity: The similarity between two BSs is defined as their negative squared Euclidean distance, i.e., $S_{m,n}=-||l_m-l_n||^2, m,n\in\{1,\cdots,N\}, m\neq n$, where l_n denotes the location of the n-th BS, and a larger $S_{m,n}$ indicates the preference of BS B_m to choose B_n as the center. Here, we set $S_{n,n}=\max_m S_{m,n}$ to denote the greatest preference of BS B_n to choose itself as center.

(II) Iteratively updating: Then, the value of R and A at each iteration are respectively computed as $R_{m,n}=S_{m,n}-\max_{n'\neq n}\{S_{m,n'}+A_{m,n'}\}$, $A_{m,n}=\min\{0,R_{n,n}+\sum_{m'\neq m,n}\max\{0,R_{m',n}\}\}$, and $A_{n,n}=\sum_{m'\neq m}\max\{0,R_{m',n}\}$, where the initial value of $A_{m,n}$ is set as zero, and a damping factor $\lambda\in[0,1]$ is also adopted to smooth the updating process [12].

(III) Obtaining cluster centers: When the value of $\arg\max_n\{A_{m,n}+R_{m,n}\},\forall m\in\mathcal{N}$, does not change, we obtain the cluster center set as

$$\mathcal{C}_{ap}=\left\{\arg\max_n\{A_{m,n}+R_{m,n}\},\forall m\in\mathcal{N}\right\}. \tag{14}$$

Remark 3.1: The cluster center set $\mathcal{C}_{ap}$ given in (14) remains during T time slots, and thus $\mathcal{K}(t)$ is replaced by $\mathcal{K}$ in the sequel. Since the size of each cluster is still changed over time, the dimension of $\mathbf{w}_{k^n}^{\mathrm{DL}}(t)$ varies over time. Problem (10)-(12) cannot been solved by conventional DRL as discussed in Remark 2.1.

B. Multi-agent DRL

To solve the above difficulties, we first treat each small cell as one agent, and define the reward for each small cell as the reward of singleton plus the extra rewards for clustering with others in the same cluster [13], i.e.,

$$r_n(t)=\sum_{k\in\mathcal{K}}\mathbb{I}_{\{c_n(t)=k\}}\iota_n^k(t)(v_{\mathcal{C}_k(t)}-\sum_{m\in\mathcal{C}_k(t)}v_{\{m\}})+v_{\{n\}}, \tag{15}$$

where $\mathbb{I}_{\{c_n(t)=k\}}=1$ indicate small cell n being in $\mathcal{C}_k(t)$, $v_{\{n\}}$ is the reward of singleton, and $\iota_n^k(t)=\frac{v_{\{n\}}}{\sum_{m\in\mathcal{C}_k(t)}v_{\{m\}}}$ measures the ratio contribution of small cell n for forming cluster $\mathcal{C}_k(t)$ with other members in $\mathcal{C}_k(t)$.

Proposition 3.1: From (9) and (15), it is derived that the sum rewards of all small cells in the considered system are equal to the sum rewards of all clusters i.e., $\sum_{n\in\mathcal{N}}r_n(t)=\sum_{k\in\mathcal{K}}v_{\mathcal{C}_k(t)}$.

Proof: : Proposition 3.1 can be easily proved by combining with (9) and (15), which is omitted due to paper limitation.

Then, problem (10)-(13) is transformed to maximize the average sum rewards of N small cells, i.e.,

$$\max_{\{\mathcal{P},\mathcal{W},\mathcal{E}\}} \lim_{T\to\infty}\frac{1}{T}\sum_{t\in\mathcal{T}}\gamma^{t-1}\sum_{n\in\mathcal{N}} r_n(t) \tag{16}$$

$$\text{s.t.} \quad 0\le p_n^{\mathrm{UL}}(t)\le P_{\mathrm{UL}}, \tag{17}$$

$$0\le p_n^{\mathrm{DL}}(t)\le P_{\mathrm{DL}},\forall t,n, \tag{18}$$

$$\mathbf{w}_n^{\mathrm{H}}(t)\mathbf{w}_n(t)=1, \tag{19}$$

$$\sum_{k\in\mathcal{K}}\mathbb{I}_{\{c_n(t)=k\}}=1,\forall t,n, \tag{20}$$

where $p_n^{\mathrm{UL}}(t)$ and $p_n^{\mathrm{DL}}(t)$ are from $p_{k_n}^{\mathrm{UL}}(t)$ and $p_{k_n}^{\mathrm{DL}}(t)$ by removing the subscript k; $\mathbf{w}_n(t)\in\mathbb{C}^{N\times 1}$ is the transmission weight at n-th BS for N DUs with $\mathbf{w}_n^{\mathrm{H}}(t)\mathbf{w}_n(t)=1$, and its m-th component satisfies $w_{n,m}(t)\sqrt{p_m^{\mathrm{DL}}(t)}=w_{k_m,k_n}^{\mathrm{DL}}(t), m,n\in\mathcal{C}_k(t), k\in\mathcal{C}_{ap}$, with $w_{k_m,k_n}^{\mathrm{DL}}(t)$ being the n-th component of the transmission parameter $\mathbf{w}_{k_m}^{\mathrm{DL}}(t)$ from the VBSE of cluster $\mathcal{C}_k(t)$ to the k_m-th DU in problem (10)-(12); $\mathcal{P}=\{p_n^{\mathrm{UL}}(t),p_n^{\mathrm{DL}}(t)\}_{n\in\mathcal{N},t\in\mathcal{T}}$ denotes the set of transmit power in all uplinks and downlinks during T time slots; $\mathcal{W}=\{\mathbf{w}_n(t)\}_{n\in\mathcal{N},t\in\mathcal{T}}$ is the set of transmission parameters from all BSs to DUs during T times; and

$\mathcal{E} = \{c_n(t)\}_{n\notin\mathcal{N},t\in\mathcal{T}}$ represents the set of clustering during T time slots.

Next, a multi-agent DRL with constant number of agents and dimension of design variable is proposed to solve problem (16)-(20). The actions and states are defined as follows.

Actions: The action at each time slot consists of two parts: clustering for non-center agents and resource allocation, where each agent chooses the power values p_n^{UL} and p_n^{DL} for the uplink and downlink transmissions, as well as the downlink transmission parameter vector $\mathbf{w}_n \in \mathbb{C}^{N\times 1}$. Since the deep neural network cannot cope with the complex number, each component $w_{n,m}$ in $\mathbf{w}_n$ is expressed as $w_{n,m} = \bar{w}_{n,m}\exp(j\vartheta_{n,m})$. Then, the action is defined as

$$a_n(t) = \left\{ \begin{array}{l} c_n(t), p_n^{\mathrm{UL}}(t), p_n^{\mathrm{DL}}(t), \bar{w}_{n,1}(t), \\ \cdots, \bar{w}_{n,N}(t), \vartheta_{n,1}(t), \cdots, \vartheta_{n,N}(t) \end{array} \right\}, \quad (21)$$

where $c_n(t) \in \mathcal{C}_{ap}$ as given in (14), $p_n^{\mathrm{UL}} \in [0, P_{\mathrm{UL}}], p_n^{\mathrm{DL}} \in [0, P_{\mathrm{DL}}], \vartheta_{n,m} \in [0, 2\pi]$, and we myopically let $\bar{w}_{n,m} \in [0,1]$[3]. It is derived that $c_n(t) = n, \forall n \in \mathcal{C}_{ap}$.

States: In the considered system, the local observation s_n of agent n is regarded as its state, which consists of its local channel condition, the members of forming the current cluster, and the previous action, i.e.,

$$s_n(t) = \left\{\mathcal{H}_n(t), \boldsymbol{c}_n^{hot}(t-1), a_n(t-1)\right\}, \quad (22)$$

where $\boldsymbol{c}_n^{hot} \in \mathbb{R}^{N\times 1}$ is obtained via an one-hot generator based on the observable members in the up-to-date cluster. The difference between our framework and conventional DRL is that the observed state about the cluster pattern at each agent will be updated after clustering, i.e., $\boldsymbol{c}_n^{hot}(t-1) \to \boldsymbol{c}_n^{hot}(t)$.

C. Clustering and Resource Allocation

Based on the above setup, a multi-agent DRL with each small cell as an agent is employed for the SD UDNs with N small cells, where the agents distributively execute the actions based on their local observations, and a centralized controller is employed to train the neural network via collecting the history experiences from all agents. Each time slot is divided into two successive phases, i.e., clustering and resource allocation phases. Multi-agent DQN is applied to instruct the non-center agents to choose one cluster for the discrete action space, while MADDPG algorithm equipped with N pairs of actor-critic networks is introduced to deal with the continuous action space for the resource allocation.

1) Execution: The action execution processes for the two phases are illustrated as follows.

- For the first phase, each non-center agent $n, n \notin \mathcal{C}_{ap}$, select the cluster center $c_n(t)$ as a function of state $s_n(t)$, which is denoted as $s_n^c(t)$ during the clustering phase in the sequel, with the help of DQN. Then, ϵ-greedy method is adopted to choose the action, i.e., choosing the action $\arg\max_{a_n^c} \bar{Q}\left(s_n^c, a_n^c; \theta'\right)$ with the probability of $1-\epsilon$, and randomly selecting other actions with probability ϵ. $\bar{Q}$ is a copied version of the DQN at agent n with parameters θ'. When the clustering is finished, each agent can observe the members in the same cluster. Thus, we update $\boldsymbol{c}_n^{hot}(t-1)$ as $\boldsymbol{c}_n^{hot}(t) = [1, \cdots, 1, 0, \cdots, 0]^T$, if the first N_k agents are in the k-th cluster.
- For the second phase, $\boldsymbol{c}_n^{hot}(t-1)$ has been replaced by $\boldsymbol{c}_n^{hot}(t)$ in $s_n(t)$, such that we redenote the state as $s_n^p(t)$ in the second phase. Then, each agent executes the action $a_n(t)$ as a function of $s_n^p(t)$ via actor network, i.e., $a_n(t) = A\mathrm{net}\left(s_n^p(t); \theta_n\right)$, where Anet denotes the actor network; the first term in $a_n(t)$ in (21) has already been obtained after clustering; and parameter θ_n is determined by the critic network. The critic outputs Q_n value to evaluate the quality of action $a_n(t)$ chosen by the actor.

When the execution is finished, we receive the corresponding reward $r_n(t)$, and then the environment turns to next state $s_n(t+1)$.

2) Training: The training DQN and actor-critic networks are deployed at a centralized controller to ease implementation and to improve stability. After execution, the experiences $e_n(t) = \left(s_n(t), a_n(t), r_n(t), s_n(t+1)\right), \forall n \in \mathcal{N}$, will be stored into a memory buffer $\mathcal{D}$ with length M. The trainer randomly selects a mini-batch of $L \ll M$ experiences from $\mathcal{D}$ that store all agents' experiences to train the networks.

- In the first phase, the centralized training network trains a single DQN by using the experiences gathered from all agents. DQN learns the action-value function corresponding to the optimal policy by minimizing the loss

$$\mathcal{L}\left(\theta\right) = \mathbb{E}_{\boldsymbol{s}^c, \boldsymbol{c}, r^c, \boldsymbol{s}^{c\prime}\sim\mathcal{D}}\left[\left(Q^{\boldsymbol{\pi}}\left(\boldsymbol{s}^c, \boldsymbol{c}; \theta\right) - y\right)^2\right], \quad (23)$$

  where $\boldsymbol{s}^c = \{s_1^c, \cdots, s_n^c\}$, $\boldsymbol{c} = \{c_1, \cdots, c_n\}$, and $y = r_n^c + \gamma \max_{c_n'} \bar{Q}^{\boldsymbol{\pi}'}(\boldsymbol{s}^{c\prime}, \boldsymbol{c}'; \theta')$. Here, we define $r_n^c = \Pr(c_n) r_n(a_n, s_n)$, and use the empirical probability over the mini-batch L to approximately describe the value of $\Pr(c_n)$. $\bar{Q}^{\boldsymbol{\pi}'}$ is a target Q function whose parameters θ' periodically updated with the most recent θ, i.e., $\theta' \leftarrow (1-\tau)\theta + \tau\theta'$, where $\tau \ll 1$ is the update factor.
- For the second phase, each agent equips with one pair of actor-critic networks, and both of them need to be trained at the central controller. Note that the actor training needs to use the global experience from all agents. Thus, the actor is trained by minimize the gradient given in (24), where $\boldsymbol{a} = \{a_1, \ldots, a_N\}$, $\boldsymbol{s}^p = \{s_1^p, \ldots, s_N^p\}$. Here, $Q_n^{\boldsymbol{\mu}}\left(\boldsymbol{s}, \boldsymbol{a}\right)$ is the output of critic network, which is used to evaluate the action chosen by the actor. The critic

$$\nabla_{\theta_n} J\left(\boldsymbol{\mu}_n\right) = \mathbb{E}_{\boldsymbol{s}^p, \boldsymbol{a}\sim\mathcal{D}}\left[\nabla_{a_n} Q_n^{\boldsymbol{\mu}}\left(\boldsymbol{s}^p, \boldsymbol{a}\right)\big|_{a_n=\boldsymbol{\mu}_n(s_n^p)}\right] \nabla_{\theta_n}\boldsymbol{\mu}_n\left(a_n \mid s_n^p\right). \quad (24)$$

[3]It will be normalized to satisfy the condition of $\mathbf{w}_n^{\mathrm{H}}(t)\mathbf{w}_n(t) = 1$ in simulations.

network is trained by maximizing the loss function, i.e.,

$$\mathcal{L}(\theta_n) = \mathbb{E}_{\boldsymbol{s}^p, \boldsymbol{a}, \boldsymbol{r}, \boldsymbol{s}^{p\prime} \sim \mathcal{D}} \left[\left(Q_n^{\boldsymbol{\mu}}(\boldsymbol{s}^p, \boldsymbol{a}) - y \right)^2 \right], \quad (25)$$

where $y = r_n + \gamma Q_n^{\boldsymbol{\mu}'}\left(\boldsymbol{s}^{p\prime}, \boldsymbol{a}'\right)|_{a_n' = \boldsymbol{\mu}_n'(s_n^p)}$, and $\boldsymbol{\mu}' = \{\boldsymbol{\mu}_{\theta_1'}, \ldots, \boldsymbol{\mu}_{\theta_N'}\}$ is the set of target policies with parameter $\theta_n', \forall n \in \mathcal{N}$. When each actor is trained with the help of critic network, the corresponding parameters are downloaded to the target actor to support the distributed online execution, i.e., $\theta_n' \leftarrow (1-\tau)\theta_n + \tau\theta_n', \forall n \in \mathcal{N}$, where θ_n' is the parameter of the target actor-critic pair networks for the resource allocation.

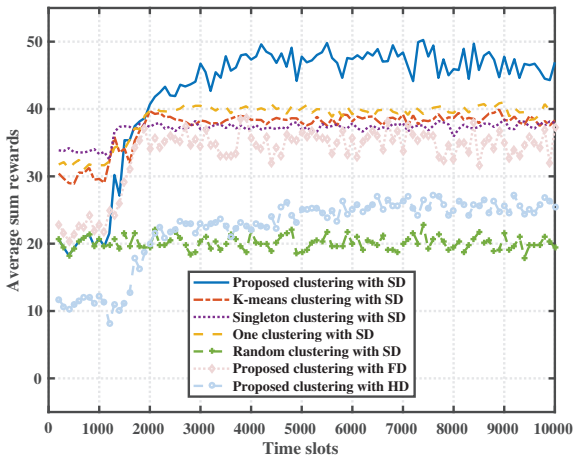

Fig. 2. Average sum rewards over time for different strategies.

IV. Simulation Results

This section present the simulation results for the proposed algorithm in SD UDN with $N = 10$. The maximum transmit power for the uplink and downlink are set as $P_{\text{UL}} = 20$dB and $P_{\text{DL}} = 25$dB, respectively. The channel coefficient is set as $h_{ij}(t) = \rho h_{ij}(t-1) + \kappa_{ij}(t)$[4], where $h_{ij}(0) = \phi_{ij}\varrho_{ij}(0)$, and $\kappa_{ij}(t) \sim \mathcal{CN}\left(0, \phi_{ij}^2 - \phi_{ij}^2\rho^2\right)$, with $\varrho_{ij}(0) \sim \mathcal{CN}(0,1)$ and $\phi_{ij} \geq 0$ being the constant large-scale fading. The path loss between transmitter m and receiver n is set as $s_{m,n}^{-3}$. The standard deviation of the log-normal shadow fading is set as 10 dB, and the AWGN power σ^2 is set as -30dB. The damping factor in the AP method is set to $\lambda = 0.7$. Moreover, the period of each time interval is set as $T_d = 70$ms, and the maximum Doppler frequency f_d is set as 10 Hz. Each neural network equips with one input layer, three hidden layers, and one output layer. We normalize each actor's output by the normalization function of Tensorflow to let $\mathbf{w}_n^{\text{H}}(t)\mathbf{w}_n(t) = 1$. Each hidden layer is set to be with 512 neurons, and rectifier linear unit (ReLU) is the activation function. The learning rate of DQN, actor, and critic networks, are set as 10^{-4}, 10^{-5}, and 10^{-4}, respectively. $M = 1000, D = 64, \gamma = 0.9$ and $q_k = 0.6, \forall k \in \mathcal{K}$, are set in the proposed DRL algorithm.

[4]The correlation coefficient is represented by $\rho = J_0(2\pi f_d T_d)$, where $J_0(\cdot)$ is the first class of zero-order Bessel function, f_d is the maximum Doppler frequency, and T_d is the length of one time slot.

Fig. 2 shows that the proposed algorithm achieves convergence within 6000 time slots. Compared with other four clustering methods, the proposed dynamic clustering performs higher average sum rewards with value about 45 to 50, while the others are below 40. Moreover, it also shows that SD outperforms both the FD and HD modes about 10 and 20, respectively, in the considered UDNs.

V. Conclusion

In this paper, a dynamic clustering and resource allocation scheme was proposed in a SD-powered UDN to maximize the average weighted sum of the network throughput and the clustering cost. With the AP method, the number of clusters and each center were determined, and the original problem was proved to equivalent to a multi-agent MDP to maximize the average reward of all small cells by treating each small cell as one agent. A multi-agent DRL algorithm was proposed to jointly implement the dynamic clustering for non-center small cells, resource allocation, and duplex mode selection. Simulation showed that the proposed algorithm can achieve a higher sum rewards under the considered scenarios.

References

[1] M. Chen and Y. Hao,"Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.

[2] L. Wang, K.-K. Wong, S. Jin, G. Zheng, and R. W. Heath,"A new look at physical layer security, caching, and wireless energy harvesting for heterogeneous ultra-dense networks,"*IEEE Commun. Mag.*, vol. 56, no. 6, pp. 49–55, June 2018.

[3] J. Lee and T. Q. S. Quek,"Hybrid full-/half-duplex system analysis in heterogeneous wireless networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 5, pp. 2883–2895, May 2015.

[4] C. Pan, H. Ren, M. Elkashlan, A. Nallanathan, and L. Hanzo,"Weighted sum-rate maximization for the ultra-dense user-centric TDD C-RAN downlink relying on imperfect CSI," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1182–1198, Feb. 2019.

[5] G. Zhang, F. Ke, H. Zhang, F. Cai, G. Long, and Z. Wang, "User access and resource allocation in full-duplex user-centric ultra-dense networks," *IEEE Trans. Veh. Tech.*, vol. 69, no. 10, pp. 12 015–12 030, Jul. 2020.

[6] Z. Li and C. Guo, "Multi-agent deep reinforcement learning based spectrum allocation for D2D underlay communications," *IEEE Trans. Veh. Tech.*, vol. 69, no. 2, pp. 1828–1840, Feb. 2020.

[7] H. Peng and X. Shen, "Multi-agent reinforcement learning based resource management in MEC- and UAV-assisted vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 131–141, Jan. 2021.

[8] Y. Xu, W. Xu, Z. Wang, J. Lin, and S. Cui, "Load balancing for ultra dense networks: A deep reinforcement learning-based approach," *IEEE Internet of Things J.*, vol. 6, no. 6, pp. 9399–9412, Aug. 2019.

[9] T. M. Cover, J. A. Thomas, "Elements of Information Theory," New York:Wiley, 1991.

[10] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, Oct. 2019.

[11] D. Wang, J. Huang, C. Huang, and G. Shi, "Interference channels with full-duplex amplify-and-forward receiver cooperations," *IEEE Access*, vol. 8, pp. 203 901–203 916, Jul. 2020.

[12] L. Li, C. Yang, M. E. Mkiramweni, and L. Pang, "Intelligent scheduling and power control for multimedia transmission in 5G CoMP systems: A dynamic bargaining game," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 7, pp. 1622–1631, Jul. 2019.

[13] A. Asheralieva and D. Niyato, "Hierarchical Game-Theoretic and Reinforcement Learning Framework for Computational Offloading in UAV-Enabled Mobile Edge Computing Networks With Multiple Service Providers," *IEEE Internet of Things J.*, vol. 6, no. 5, pp. 8753-8769, Oct. 2019.