Drone-Cell Trajectory Planning and Resource Allocation for Highly Mobile Networks: A Hierarchical DRL Approach

Weisen Shi, *Graduate Student Member, IEEE*, Junling Li, *Graduate Student Member, IEEE*, Huaqing Wu, *Graduate Student Member, IEEE*, Conghao Zhou, *Graduate Student Member, IEEE*, Nan Cheng, *Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Drone cell (DC) is envisioned to enable the dynamic service provisioning for radio access networks (RANs), in response to the spatial and temporal unevenness of user traffic. In this article, we propose a hierarchical deep reinforcement learning (DRL)-based multi-DC trajectory planning and resource allocation (HDRLTPRA) scheme for high-mobility users. The objective is to maximize the accumulative network throughput while satisfying user fairness, DC power consumption, and DC-to-ground link quality constraints. To address the high uncertainties of the environment, we decouple the multi-DC TPRA problem into two hierarchical subproblems, i.e., the higher level global trajectory planning (GTP) subproblem and the lower level local TPRA (LTPRA) subproblem. First, the GTP subproblem is to address trajectory planning for multiple DCs in the RAN over a long time period. To solve the subproblem, we propose a multiagent DRL-based GTP (MARL-GTP) algorithm in which the nonstationary state space caused by the multi-DC environment is addressed by the multiagent fingerprint technique. Second, based on the GTP results, each DC solves the LTPRA subproblem independently to control the movement and transmit power allocation based on the real-time user traffic variations. A deep deterministic policy gradient (DEP)-based LTPRA (DEP-LTPRA) algorithm is then proposed to solve the LTPRA subproblem. With the two algorithms addressing both subproblems at different decision granularities, the multi-DC TPRA problem can be resolved by the HDRLTPRA scheme. Simulation results show that 40% network throughput improvement can be achieved by the proposed HDRLTPRA scheme over the nonlearning-based TPRA scheme.**

***Index Terms*—Drone cell, drone-assisted radio access network (RAN), space-air-ground integration, trajectory planning.**

Manuscript received April 12, 2020; revised July 15, 2020 and August 14, 2020; accepted August 19, 2020. Date of publication August 28, 2020; date of current version June 7, 2021. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada; in part by the National Natural Science Foundation of China under Grant 91638204; and in part by the Shenzhen Institute of Artificial Intelligence and Robotics for Society. *(Corresponding author: Junling Li.)*

Weisen Shi, Huaqing Wu, Conghao Zhou, and Xuemin Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: w46shi@uwaterloo.ca; h272wu@uwaterloo.ca; c89zhou@uwaterloo.ca; sshen@uwaterloo.ca).

Junling Li is with the Shenzhen Institute of Artificial Intelligence and Robotics for Society and the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: lijunling@cuhk.edu.cn).

Nan Cheng is with the School of Telecommunication, Xidian University, Xi'an 710071, China (e-mail: dr.nan.cheng@ieee.org).

Digital Object Identifier 10.1109/JIOT.2020.3020067

I. Introduction

IN FUTURE radio access networks (RANs), ubiquitous network connectivity with guaranteed Quality of Service (QoS) is expected by users anywhere at anytime. However, highly dynamic and uneven distribution of terrestrial data traffic poses great challenges to ensure the seamless network connectivity, especially in high-mobility scenarios [1], [2]. Although densely deploying massive small cells can be a potential solution, it is inefficient and costly for RAN operators due to the high idle probability of small cells deployed for peak hours or remote areas [3]. To address the spatial and temporal traffic unevenness in a cost-efficient way, drones, *also known as* unmanned aerial vehicles (UAVs), equipped with wireless communication modules are leveraged as drone cells (DCs) to assist the future RAN. The deployment of DCs can benefit future RAN in three aspects: 1) providing high-quality DC-to-ground wireless links with higher Line-of-Sight (LoS) probability than the terrestrial base station (BS)-to-user links [4]; 2) enabling flexible deployment in response to the spatial and temporal terrestrial traffic variations [5]; and 3) executing onboard computing tasks or algorithms for terrestrial RAN [6], [7]. As a promising technology, the DC-assisted RANs (DA-RANs) have attracted increasing research attentions [8] and field tests [9]. The 3rd Generation Partnership Project (3GPP) is also initiating the standardization process of seamlessly integrating UAV into future cellular networks [10].

In DA-RAN, DCs are dynamically deployed to serve users in the uncovered areas of terrestrial RAN, and relay data between users and BSs via DC-to-BS (D2B) links. The uncovered areas include both the blind spots of terrestrial RAN's communication coverage and the bursty traffic spots (BTSs) where terrestrial RAN resources are inadequate to support the dense traffic (e.g., congested roads, a stadium with sports events, etc.). Since the DA-RAN's capability to address terrestrial traffic variations is enabled by the dynamic deployment feature of DCs, the DC trajectory planning problem, which designs flying traces and movements of DCs to serve users,

is essential for DA-RAN. Besides, given the trajectory of each DC, the resource allocation strategy of each DC should not only address the user mobility impacts but also adapt to the varying deployment locations, which cannot be addressed by conventional resource allocation schemes for statically deployed RAN. Therefore, the DC resource allocation should be jointly investigated with the DC trajectory planning, which forms the DC trajectory planning and resource allocation (TPRA) problem.

The DC TPRA problem has been studied by some pioneer works with the objectives of improving given performance metrics [11], [12], e.g., network throughput, QoS of users, etc. In these nonlearning-based works, the terrestrial users or uncovered areas are modeled as static nodes for each TPRA process. Given the fixed user locations and known traffic model, a deterministic TPRA decision for each DC, which cannot adapt itself to traffic variations, is calculated through optimization methods. The trajectory of each DC is a closed curve composed of discrete 3-D locations. The DC sequentially transverses each location and serves the associated users (covered areas) at scheduled time slots. If the user locations or traffic patterns change significantly, replanning and reallocation need to be executed. However, the deterministic trajectory and static user model is only applicable for fixed or low-mobility users with scheduled communications, e.g., data collection for massive Internet-of-Things (IoT) devices [13]. To adapt the nonstatic environment, the learning-based TPRA research leverages machine learning techniques to make TPRA decisions for DCs according to environment variations [14]. However, most of the existing works consider scenarios with single DC, fixed number of users, single BS communication coverage, or only designing the trajectory without resource allocation [15]–[19]. When focusing on a large RAN area with nonstatic users, the number of users is spatiotemporal variant due to the high user mobility. Besides, multiple DCs are required to cooperatively serve the ground users, which involves potential mutual interference among DCs. Therefore, it is still challenging to propose a learning-based TPRA scheme in the general environment with multiple DCs and the nonfixed number of high-mobility users over large RAN areas.

In this article, we investigate the multi-DC TPRA problem to serve high-mobility users (e.g., vehicular users) over large RAN areas with multiple BSs. Considering the user fairness, DC energy consumption, DC to user (D2U), and D2B communication constraints, the multi-DC TPRA problem is formulated as a constrained Markov decision process (CMDP) aiming to maximize the long-term accumulative network throughput over the large area. However, the multi-DC TPRA problem is intractable for conventional deep reinforcement learning (DRL)-based algorithms in the complex environment due to the high spatiotemporal network dynamics and the inter-DC interference. Therefore, we propose a hierarchical deep reinforcement learning (HDRL)-based TPRA (HDRLTPRA) scheme to decouple the highly complicated problem into two hierarchical subproblems. The objective of the higher level global trajectory planning (GTP) subproblem is to plan global trajectories for multiple DCs by the RAN controller to maximize the accumulated number of served users over a large area and a long time period. The global trajectory determines the sequence of areas (e.g., BTSs) served by each DC. Based on the global trajectory decision at each GTP step, the lower level local TPRA (LTPRA) subproblem is addressed by each DC independently to control real-time movement and allocate resources within the predetermined area. To solve the two subproblems in HDRLTPRA, a MARL-GTP algorithm and a deep deterministic policy gradient-based LTPRA (DEP-LTPRA) algorithm are, respectively, designed. Simulations demonstrate that 40% improvement in terms of total network throughput is achieved by the HDRLTPRA scheme in comparison with the nonlearning-based TPRA scheme. The main contributions of this work are threefold.

1) We propose an effective multi-DC HDRLTPRA scheme for high-mobility scenarios. In HDRLTPRA, the higher level MARL-GTP algorithm addresses the complexities caused by multiple DCs and long-term variations of user distributions. The lower level DEP-LTPRA algorithm addresses the real-time variations in the number and locations of served users, with the state space constrained by the output of the higher level MARL-GTP algorithm. This hierarchical DRL framework allows the HDRLTPRA to converge to suboptimal TPRA solutions with high probability.
2) To generate the global trajectories, we design the MARL-GTP algorithm which implements fully cooperative multiagent DRL (MARL) to fit the multi-DC environment. In specific, the multiagent fingerprints and prioritized experience replay (PER) methods are applied to design the hyper-parameters and neural network (NN) in the MARL-GTP algorithm to address the nonstationary environment and sparse rewards.
3) In response to the real-time user mobility, we design the DEP-LTPRA algorithm executed by each DC independently to adjust the real-time DC flying control and resource (i.e., transmit power) allocation. In the DEP-LTPRA algorithm, the DEP enables the TPRA over continuous spaces. Besides, the complexity of DEP-LTPRA's input is reduced by mathematical analyses of D2U communication, which further improves the convergence performance.

The remainder of this article is organized as follows. The related works are discussed in Section II. In Section III, we introduce the system model. Then, the multi-DC TPRA problem is formulated and decoupled into two hierarchical subproblems in Section IV. In Section V, the HDRLTPRA scheme is proposed with the higher level MARL-GTP algorithm and the lower level DEP-TPRA algorithm. Real-world scenario-based simulations are carried out in Section VI, followed by conclusions in Section VII.

II. Related Works

As the pioneer works of nonlearning-based DC TPRA research, [11] and [20] define the initial TPRA problem in which the DC periodically serves quasistatic terrestrial users through a discrete 3-D trajectory. The nonlearning-based

TPRA solution has been investigated in various scenarios, including air-ground integrated communication assistance [21] and IoT data collections [22]. To model the mobility and traffic of terrestrial users, the well-developed network capacity research for the cooperative mobile network is usually leveraged. For instance, Song *et al.* [23] in the first time derived the scaling bound on cooperative network capacity, which is recognized as the basis on the design of mobile networks, particularly for highly mobile communications. The important results founded in [24] show the tradeoff between communication and secrecy capacity, therefore to construct optimal secure network codes for cooperative mobile networks. The nonlearning-based DC TPRA is generally jointly optimized with other impact factors, such as DC power consumption [25], DC altitude and speed [26], number of DCs [13], etc. Although the discrete trajectory and quasistatic user models simplify the optimization process, the accuracy of the TPRA solutions is inevitably decreased.

To address the nonstationary environment with model-free methods, the learning-based approaches are proposed for the DC TPRA problem. Most learning-based TPRA works use the DRL framework, where the DC takes TPRA actions according to the observed environment state, and then receives reward in each step. The policy of choosing TPRA actions is updated step by step to reach the convergence, with the objective of maximizing the long-term accumulative reward. In [15], the deterministic policy gradient (DPG) learning method is leveraged in single DC trajectory planning to maximize user throughput. A DRL-based interference-aware trajectory planning scheme is proposed for single DC in [16], which achieves low D2U latency and high throughput. Lu *et al.* [17] designed a relay scheme of single DC, which integrates both Q-learning and DRL to minimize both bit error rate of relayed signal and the DC power consumption. Considering the multi-DC scenario, a decentralized DRL framework is proposed in [18] to solve multi-DC trajectory design problem based on a sense-and-send protocol. Given the power consumption constraints of DC, Liu *et al.* [19] designed a DRL-based energy-efficient trajectory planning method for fair communication coverage of multiple DCs. In [27], the DRL-based computation tasks offloading is proposed for DC assisted IoT scenario. Although various impact factors are considered by existing DRL-based TPRA research, the multi-DC environment, nonfixed user number and locations, as well as the joint TPRA of DC have never been considered simultaneously. In this work, we will propose the HDRLTPRA scheme to solve the multi-DC TPRA problem in high-mobility scenarios.

III. System Model

A. DA-RAN Scenario

Fig. 1 shows the DA-RAN scenario where two DCs embedded on rotary-wing drones are released by the DA-RAN to serve high-mobility users (e.g., vehicular users) over a large area. In this work, we consider downlink transmissions in DA-RAN where DCs relay data from BS to their served users using additional spectrum resources. Define the DA-RAN scenario as $\mathcal{G}$ whose area is large enough to contain

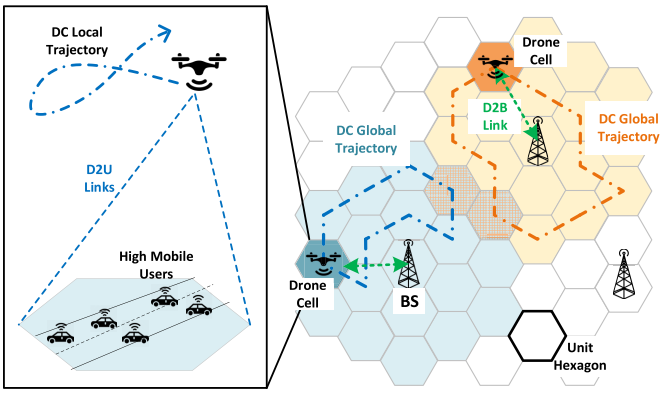


Fig. 1. System model.

multiple BSs. For example, $\mathcal{G}$ can be a university campus with its affiliated regions. In the DA-RAN scenario $\mathcal{G}$, denote $\mathcal{B} = \{b_1, b_2, \ldots, b_{|\mathcal{B}|}\}$ as the set of BSs with cardinality $|\mathcal{B}|$, and $\mathcal{D} = \{d_1, d_2, \ldots, d_{|\mathcal{D}|}\}$ as the set of DCs with cardinality $|\mathcal{D}|$. The communication coverage of each BS $b \in \mathcal{B}$ is modeled as a hexagon area with a circumscribed circle radius R_b. The blue and yellow areas in the right side of Fig. 1 represent the coverage of two BSs, respectively. The communication coverage of each DC $d \in \mathcal{D}$ is also a hexagon area with a circumscribed circle radius R_d, as shown in the left side of Fig. 1. To simplify the environment for DC trajectory planning, we evenly divided the whole scenario into a hexagon mesh where the size of each unit hexagon (unit) equals one DC's communication coverage. The global trajectory of each DC is therefore defined as the sequence of units served by it. Within each unit among one DC's global trajectory, the DC can dynamically adjust its movement according to the ground traffic variation in the unit, which forms the local trajectory of the DC. To avoid possible collisions and interference among DCs, we define that each unit only allows at most one DC to fly over it. Denoting each unit as g, we can redefine $\mathcal{G}$ as a set $\mathcal{G} = \{g_1, g_2, \ldots, g_{|\mathcal{G}|}\}$ with the cardinality $|\mathcal{G}|$. The communication coverage of BS b is further modeled as a set $\mathcal{G}_b = \{g_b \ldots\}$ containing multiple units covered by BS b. Note that $\mathcal{G}_b \cap \mathcal{G}_{\bar{b}} \geq \emptyset \ \forall b \neq \bar{b}$ due to the coverage overlapping of adjacent BSs.

Because of the high-mobility feature, the total numbers and locations of users in the whole scenario $\mathcal{G}$ are random variables that change over time. Define the decision step t with length δ_t, during which both DCs and high-mobility users can only move within sufficiently small ranges with negligible effects on the network performance. The locations of each DC d and each user u at step t are represented by $z_d(t) = (x_d(t), y_d(t), h_d(t))$ and $(x_u(t), y_u(t))$, respectively, where (x_d, y_d, h_d) and (x_u, y_u) are 3-D and 2-D Cartesian coordinates, respectively. All users served by DC d at step t form a set $\mathcal{U}_d(t)$ with the cardinality (number of users) $|\mathcal{U}_d(t)|$, and all users in unit g at step t is represented by the set $\mathcal{U}_g(t)$ with the cardinality $|\mathcal{U}_g(t)|$. Since we mainly address users' high-mobility issue in this work, we assume all users are running the same type of services with homogeneous downlink data traffic patterns and bandwidth requirements.

B. DC Communication Model

The state-of-the-art D2U and D2B channel models are used to indicate the high LoS probabilities of both D2U and D2B links. According to [28], the average D2U path loss (dB) is calculated by

$$P_{du}^{L}(r_{du}(t), h_d(t)) = 20\log\left(\frac{4\pi f_c}{c}\sqrt{h_d(t)^2 + r_{du}(t)^2}\right) + \mathrm{Pr}_{\mathrm{los}}\eta_{\mathrm{los}} + (1-\mathrm{Pr}_{\mathrm{los}})\eta_{\mathrm{nlos}} \quad (1)$$

where $r_{du}(t)$ (m), $h_d(t)$ (m), f_c (Hz), and c (m/s) are D2U horizontal distance, DC flying height, carrier frequency, and speed of light, respectively. Let $\mathrm{Pr}_{\mathrm{los}}$ denote the LoS probability, calculated from

$$\mathrm{Pr}_{\mathrm{los}} = \frac{1}{1 + a\exp\left(-b\arctan\left(\frac{h_d(t)}{r_{du}(t)}\right) + ab\right)}. \quad (2)$$

In (1) and (2), a, b, η_{los}, and η_{nlos} are all environment-based constants.

The D2B channel is naturally modeled as LoS channel with environment-based offsets [29]. The D2B path loss (dB) is calculated by

$$P_{db}^{L}(r_{db}(t), h_d(t)) = 10\alpha\log(r_{db}(t)) + \eta_0 + A\left(\arctan\left(\frac{h_d(t)}{r_{db}(t)}\right) - \theta_0\right) \times e^{\left(\frac{\theta_0 - \arctan\left(\frac{h_d(t)}{r_{db}(t)}\right)}{B}\right)} \quad (3)$$

where $r_{db}(t)$ represents the D2B horizontal distance, A the excess path-loss scalar, α the terrestrial path-loss exponent, θ_0 the angle offset, η_0 the excess path-loss offset, and B the angle scalar. The 850-MHz LTE band is used in (3) [29]. Both (1) and (3) are large-scale path-loss models. Since this work mainly investigates the impact of user mobility on the environment complexity, the small-scale channel shadowing and fading are not involved for analysis simplicity.

To prevent interference between DCs, we assume all DCs are assigned with orthogonal spectrum resources with the same total bandwidth B_D. Since the users are homogeneous in terms of their bandwidth requirement, to ensure user fairness, the B_D of each DC is evenly shared by all its associated users with a maximal per user bandwidth constraint b_U. Denote DC d's downlink transmit power to user u at step t by $P_{du}(t)$, the downlink throughput of user u at step t is calculated by

$$c_{du}(t) = \begin{cases} b_U\log_2\left(1 + \frac{P_{du}(t)\beta_{du}^{-1}(r_{du}(t),h_d(t))}{\sigma_0 b_U}\right), & b_U \le \frac{B_D}{|\mathcal{U}_d(t)|} \\ \frac{B_D}{|\mathcal{U}_d(t)|}\log_2\left(1 + \frac{P_{du}(t)\beta_{du}^{-1}(r_{du}(t),h_d(t))}{\sigma_0 B_D/|\mathcal{U}_d(t)|}\right), & \text{otherwise} \end{cases} \quad (4)$$

where $\beta_{du}(r_{du}(t), h_d(t)) = 10^{P_{du}^{L}(r_{du}(t),h_d(t))/10}$ is the power attenuation coefficient and σ_0 is the spectral density of noise power.

Although the total bandwidth resource is assumed to be evenly shared by users of the DC, the achieved throughput of each user is still uneven due to different D2U 3-D distances and different transmit power levels allocated by the TPRA strategies. We apply Jain's fairness index [30] to measure the throughput fairness between all users served by one DC d at step t

$$m_d(t) = \frac{\left(\sum_{u\in\mathcal{U}_d(t)} c_{du}(t)\right)^2}{|\mathcal{U}_d(t)|\sum_{u\in\mathcal{U}_d(t)} c_{du}(t)^2} \quad (5)$$

where $m_d(t) \in [1/n, 1]$. The fairness level between users increases as the fairness index $m_d(t)$ increases.

C. DC Power Consumption Model

The limited total battery energy of each DC E_d is mainly consumed by three parts, i.e., the computation energy, data transmission energy, and propulsion energy. The computation energy enables the signal processing and computation functions on DC, which is relatively much smaller than the communication and propulsion energy. Without loss of generality, we ignore the computation energy consumption in this work. Denote $P_d(t)$ as the total transmit power of DC d at step t

$$P_d(t) = \sum_{u\in\mathcal{U}_d(t)} P_{du}(t). \quad (6)$$

The propulsion power energy is used to keep the DC aloft and adjust the movements. For a rotary-wing DC flying with speed $v_d(t)$ at step t, the propulsion power consumption can be modeled as [31]

$$P_d^{\mathrm{prop}}(t) = P_b\left(1 + \frac{3v_d(t)^2}{{V_{\mathrm{tip}}}^2}\right) + \frac{P_i V_h}{v_d(t)} + \frac{D_0 S_0 \rho A_0 v_d(t)^3}{2} \quad (7)$$

where P_b and P_i are DC's blade profile power and induced power in hovering state, respectively, V_{tip} denotes the tip speed of the rotor blade, V_h is the mean rotor induced velocity in hovering state, D_0, S_0, ρ, and A_0 are the fuselage drag ratio, rotor solidity, air density, and rotor disc area, respectively.

Define the service endurance T_d as the DC's continuously flying time from fully charged state to energy depletion, we have

$$\sum_{t=1}^{T_d}\left(P_d(t) + P_d^{\mathrm{prop}}(t)\right)\delta_t \le E_d. \quad (8)$$

For TPRA task conducted over long time period $T > T_d$, it is impossible for the DC to keep active with its limited battery capacity. In this article, we assume all DCs can fly back to the associated BSs to charge their batteries, with the same charging speed denoted by p_{crg} Joule per step t. The DCs in charging state cannot serve any users. In the following sections, we use $e_d(t)$ to denote the remaining battery energy of DC d at step t:

$$e_d(t) = E_d + \sum_{\tau\le t}^{t}\left[p_{\mathrm{crg}}\zeta(\tau) - \left(P_d(\tau) + P_d^{\mathrm{prop}}(\tau)\right)\delta_t(1-\zeta(\tau))\right]. \quad (9)$$

$\zeta(\tau) = 1$ when DC d is charging at step τ, otherwise $\zeta(\tau) = 0$.

IV. Problem Formulation

In this section, the multi-DC TPRA problem is formulated first, then decoupled into the higher level GTP subproblem and the lower level LTPRA subproblem.

A. Multi-DC TPRA Problem

The objective of the multi-DC TPRA problem is maximizing the accumulative network throughput over time period T, by choosing the appropriate TPRA decisions for each DC d at each step t to serve high-mobility users in DA-RAN scenario $\mathcal{G}$. Since the highly dynamic and uncertain user distributions over the scenario $\mathcal{G}$ can be assumed to evolve in an ergodic way, we can model the multi-DC TPRA problem as an MDP.

A typical MDP is denoted by a tuple $(\boldsymbol{\mathcal{S}}, \mathcal{A}, W, P)$, in which $\boldsymbol{\mathcal{S}}$ is the state space, $\mathcal{A}$ is action space, $W \coloneqq \boldsymbol{\mathcal{S}} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $P \coloneqq \boldsymbol{\mathcal{S}} \times \mathcal{A} \times \boldsymbol{\mathcal{S}} \to \mathbb{R}$ is state transition probability. Denote the user distribution status in $\mathcal{G}$ at step t by $U(t)$, we define the system state at step t as $\mathbf{S}(t) = [U(t), \mathbf{Z}(t), \mathbf{E}(t)]$. $\mathbf{Z}(t)$ is the set of all DCs' locations at step t, and $\mathbf{E}(t)$ is the set of all DCs' remaining energy at step t. We denote all DCs' trajectory planning decisions and resource allocation decisions at each step t by $\mathbf{A}_z(t)$ and $\mathbf{A}_r(t)$, respectively. The system action at step t can be represented by $\mathbf{A}(t) = (\mathbf{A}_z(t), \mathbf{A}_r(t))$. The state transitions from step t to $t+1$ are updated by three components. The first component is the user distribution status change $U(t) \to U(t+1)$, which depends on the highly dynamic and uncertain environment. The second component is all DCs' trajectories updates $\mathbf{Z}(t+1) = \mathbf{Z}(t) + \mathbf{A}_z(t)$ by taking action $\mathbf{A}_z(t)$. The third component is the remaining energy updates depend on both TPRA actions. Define the time-invariant stationary policy mapping from any state $\mathbf{S} \in \boldsymbol{\mathcal{S}}$ to any action $\mathbf{A} \in \boldsymbol{\mathcal{A}}$ as $\boldsymbol{\Pi}(\mathbf{S}, \mathbf{A})$. We aim to find the optimal policy $\boldsymbol{\Pi}^*$ to maximize the long-term expectation of average network throughput. Therefore, we formulate the multi-DC TPRA problem, which can be regarded as a CMDP, as follows:

$$\max_{\boldsymbol{\Pi}} \ \mathbb{E}\left[\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{d=1}^{D} \sum_{u}^{\mathcal{U}_d(t)} c_{du}(\mathbf{S}(t)) \middle| \boldsymbol{\Pi}\right] \tag{10}$$

$$\text{s.t.} \ \ e_d(t) \geq 0 \quad \forall d, t \tag{10a}$$

$$m_d(t) = 1 \quad \forall d, t \tag{10b}$$

$$P^L_{\mathrm{DB}}(r_{db}(t), h_d(t)) \leq \gamma_{\mathrm{DB}} \quad \forall d, t \tag{10c}$$

$$\mathcal{U}_d(t) \cap \mathcal{U}_{\bar{d}}(t) = \emptyset \quad \forall d \neq \bar{d}, t \tag{10d}$$

$$v_d(t) < V_{\max} \quad \forall d, t. \tag{10e}$$

In (10), γ_{DB} denotes the maximal allowed D2B large-scale path loss which ensures the D2B link quality. Equation (10a) is the remaining energy constraint of each DC. Equations (10b) and (10c) are user fairness constraint and D2B link quality constraint for each DC at each step t, respectively. Equation (10d) ensures multiple DCs do not have coverage overlap to improve DC utilization efficiency. Equation (10e) is the DCs' maximal flying speed constraint.

Considering massive high-mobility users over the large area $\mathcal{G}$, the size of state space $\mathcal{S}$ for $\boldsymbol{\Pi}(\mathbf{S}, \mathbf{A})$ is tremendous, which is infeasible for simple DRL-based algorithm to solve. To address the complexity, the hierarchical DRL framework is leveraged which decouples the problem into multiple subproblems with smaller state space $\boldsymbol{\mathcal{S}} \subset \mathcal{S}$, then solves the whole problem by solving all subproblems iteratively. In this article, we decouple the multi-DC TPRA problem into two hierarchical subproblems, i.e., the multi-DC GTP subproblem, and the single DC LTPRA subproblem.

B. Multi-DC GTP Subproblem

Note the fact that although the real-time user distribution varies dynamically over different time steps, the number of users within each unit $|\mathcal{U}_g(t)|$ changes little between adjacent steps, since most users' moving range within one step t cannot exceed the unit area. On the other hand, the $|\mathcal{U}_g(t)|$ varies smoothly over a long time period T (e.g., the number of vehicles within one road segment at peak hours and normal hours over one day time). These long-term trends of different units should be learned by the trajectory planning algorithm to determine global trajectories for DCs.

In this work, we denote each DC d's global trajectory by $\mathbf{z}^r_d(t_r)$. The global trajectory $\mathbf{z}^r_d(t_r)$ determines the unit g at which the DC is designed to fly over and serve users at GTP step t_r. The length of each GTP step δ_r is longer than δ_t, since the user number statistic in each unit varies slower than the exact locations of users. Define the average number of users in unit $g \in \mathcal{G}$ over GTP step t_r as $\overline{U_g}(t_r)$, the system state for multi-DC GTP subproblem is represented by $\mathbf{s}^r(t_r) = [\overline{\mathcal{U}}_g(t_r), \mathbf{z}^r(t_r), \mathbf{E}(t_r)]$. $\overline{\mathcal{U}}_g(t_r)$ is the set of all $\overline{U_g}(t_r)$, $\mathbf{z}^r(t_r)$ is the set of all DCs' global trajectory locations (assigned units) at GTP step t_r, and $\mathbf{E}(t_r)$ is the set of all DCs' remaining battery energy at GTP step t_r. Although the statistic data $\overline{U_g}(t_r)$ average the random bursts of user traffic over short step t, the long-term uncertainties and dynamics of the user traffic are kept in $\boldsymbol{\mathcal{S}}^{\mathbf{r}}$, which still requires the DRL-based algorithm to address. In the next GTP step $(t_r + 1)$, we define that the DC can only move to one of the six neighbor units or remain in its current unit. Therefore, the system action $\mathbf{a}^r(t_r) \in \boldsymbol{\mathcal{A}}^{\mathbf{r}}$ is composed of D units selected from the seven potential units of each DC, respectively. The state transitions are updated by the mean user number variations $\overline{\mathcal{U}}_g(t_r) \to \overline{\mathcal{U}}_g(t_r + 1)$, the DC location update $\mathbf{z}^r(t_r + 1) = \mathbf{z}^r(t_r) + \mathbf{a}^r(t_r)$, as well as the remaining energy update by reducing E^r_δ Joule at each t_r. The E^r_δ is calculated by $(E_d \delta_r / T_d)$, which constrains the total energy consumption (consumed by LTPRA actions) of each DC within one GTP step period. Since we assume all users are homogeneous with a constant data traffic rate, the objective of (10), is equivalent to maximizing the long-term expectation of the average number of served users. Let $\overline{U_d}(t_r)$ denote the average number of users served by DC d at GTP step t_r. Then, the multi-DC GTP subproblem can be formulated as a CMDP

$$\max_{\pi^r} \ \mathbb{E}\left[\lim_{T\to\infty} \frac{1}{T} \sum_{t_r=1}^{T} \sum_{d=1}^{D} \overline{U_d}(t_r) \middle| \pi^r\right] \tag{11}$$

$$\text{s.t.} \ \ e^r_d(t_r) \geq 0 \quad \forall d, t_r \tag{11a}$$

$$g(d, t_r) \in \mathcal{G}_b \quad \forall d, t_r \tag{11b}$$

$$g(d, t_r) \neq g\big(\bar{d}, t_r\big) \quad \forall g_k, d \neq \bar{d}, t_r \tag{11c}$$

where π^r is the stationary policy mapping from $\mathbf{s}^r \in \boldsymbol{\mathcal{s}}^r$ to $\mathbf{a}^r \in \boldsymbol{\mathcal{a}}^r$. $g(d, t_r)$ denotes the unit g served by DC d at GTP step t_r. Equation (11a) indicates the nonnegative remaining battery energy. For $T \geq T_d$, (11a) can be guaranteed by charging the DC at its associated BS periodically. Equation (11b) represents that the DC can only serve units within the communication of its associated BS. For (11c), each unit cannot be served by multiple DC simultaneously, which prevents the potential collisions and interference.

Note that the DC is able to adapt its antenna or beam directions to keep covering all users in $g_k(t_r)$ during each GTP step t_r. Given the fixed communication coverage (one unit) at each GTP step t_r, the optimal DC flying height $h_d^{\text{opt}}(t_r)$ maximizing D2U throughput can be calculated according to [13]. Therefore, we assume that the optimal flying height of DC $h_d^{\text{opt}}(t_r)$ within each GTP step t_r is known *a priori*, which simplifies the state and action space for the single DC LTPRA subproblem.

C. Single DC LTPRA Subproblem

Given the GTP result $g(d, t_r)$ at each GTP step t_r, each DC d independently solves the single DC TPRA subproblem within its communication coverage $g(d, t_r)$ at each step t. Although the decision step is as short as t to capture the real-time user mobility patterns, the complexity of state space $\boldsymbol{\mathcal{s}}_d^l$ and action space $\boldsymbol{\mathcal{a}}_d^l$ are both reduced due to the small serving area (one unit) and single DC decision (no joint action space of multiple DCs). For each DC d, the state at step t is represented by $\mathbf{s}_d^l(t) = [\mathcal{U}_d(t), \mathbf{z}_d^l(t), e_d^l(t)]$, where $\mathcal{U}_d(t)$ is the status of served users, $\mathbf{z}_d^l(t)$ is the DC location, and $e_d^l(t) \in [0, E_\delta^r]$ is the remaining energy. The action at step t $\mathbf{a}_d^l(t)$ is composed of trajectory planning action and resource allocation action. Therefore, the state transitions from t to $t+1$ are determined by the user distribution variations, trajectory planning actions, and energy consumption by TPRA actions. With the objective of maximizing the expectation of average DC throughput, we formulate the single DC LTPRA subproblem for each DC d as follows:

$$\max_{\pi_d^l} \ \mathbb{E}\left[\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{u}^{\mathcal{U}_d(t)} c_{du}\left(\mathbf{s}_d^l(t), \mathbf{a}_d^l(t)\right) \middle| \pi_d^l \right] \tag{12}$$

$$\text{s.t.} \ \ (10a)-(10c), \ (10e) \tag{12a}$$

π_d^l is the LTPRA stationary policy maps $\mathbf{s}_d^l \in \boldsymbol{\mathcal{s}}_d^l$ to $\mathbf{a}_d^l \in \boldsymbol{\mathcal{a}}_d^l$.

V. Hierarchical DRL-Based Trajectory Planning and Resource Allocation Scheme

In this section, we first propose the HDRLTPRA scheme to decouple the multi-DC TPRA problem in a hierarchical way. Then, the MARL-GTP algorithm for higher level subproblem, and the DEP-LTPRA algorithm for lower level subproblem are introduced, respectively.

A. HDRLTPRA Scheme

Considering the unknown transition probabilities, and the large state spaces due to highly dynamic and uncertain user mobility, DRL-based algorithms are required to solve the two subproblems. To integrate DRL algorithms for different subproblems to solve the multi-DC TPRA problem, we apply the hierarchical DRL framework [32] and propose the HDRLTPRA scheme, as shown in Fig. 2.

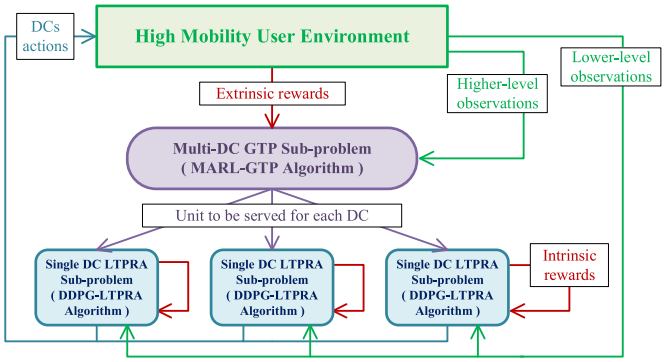


Fig. 2. Architecture of HDRLTPRA scheme.

The proposed HDRLTPRA scheme has two levels. The MARL-GTP algorithm is executed on higher level which addresses the multi-DC GTP subproblem. The higher level agent interacts with the high-mobility environment in a low-frequency rate, and observes the statistic user data, i.e., average user number within each unit in GTP step t_r, as the input state. The extrinsic reward received by higher level agent is the average number of users served by all DCs in the current GTP step. The output actions of higher level agent are units to be served by every DC in the next GTP step.

The lower level of HDRLTPRA is composed of multiple agents executing the DEP-LTPRA algorithms to address the single DC LTPRA subproblem. Each DC acts as one agent and executes the DEP-LTPRA algorithm independently. The GTP result received from higher level agent constrains the movement and serving a range of the lower level agent within one unit, which allows the DEP-LTPRA algorithm to determine TPRA decisions in response to the real-time environment variations in the local unit. The intrinsic reward is obtained by each DC independently from interacting with the environment, which is defined as the total achieved throughput by DC over the currently served unit.

The higher level and lower level agents are designed to be "offline trained" first, then "online operated" by the RAN or DCs in real-time after reaching convergence. The training phase can only be stopped after all agents in higher and lower levels reach their convergence. In real implementations, the higher level agent is implemented in the central controller of the DA-RAN to collect global data over the whole scenario and interact with multiple DCs. The benefits of this centralized solution are threefold: first, considering the limited computing resources on each drone cell, shifting the GTP tasks to the RAN side can save more computing resources for running the DEP-LTPRA algorithm. Second, the centralized MARL-GTP algorithm can prevent complex data exchange and communication processes among physical drone cells, which simplifies the system and reduces delay caused by interdrone-cell data exchanges. Third, with the global user

traffic collected by the RAN server over the whole DA-RAN scenario, the performance of the MARL-GTP algorithm can be promoted. Each lower level agent is implemented on individual DC and receives higher level GTP results via D2B communications. To increase the ubiquity of the trained lower level NN and simplify the system design, the DEP-LTPRA algorithm just trains one NN suitable for all units, and implements the NN to all DCs.

B. MARL-GTP Algorithm

Given the discrete action spaces of the higher level GTP subproblem, the deep Q-network (DQN) can be used to solve the subproblem with fast convergence. However, since the GTP subproblem considers multiple DCs, the dimension (size) of the joint action space $|\boldsymbol{\mathscr{a}}^r| = 7^D$ increases exponentially as DC number D increases. To prevent the "curse of dimensionality" for action space in the multi-DC environment, the MARL is used in the MARL-GTP algorithm. MARL has been widely applied to solve the high-dimensional joint action space issue in existing works [33], [34]. By implementing separate DRL algorithms in each agent, the global convergence is approximated through each agent's learning based on its own observation and the intercommunications between agents. In the MARL-GTP algorithm, we apply the fully cooperative MARL that all agents jointly maximize the global accumulative reward. Each DC is implemented with an identical DQN as the learning agent. Detail designs of the MARL-GTP algorithm are listed as follows.

1) Action Design: The joint action space of multiple DCs is decoupled into multiple local action spaces for different agents. The local action for each agent is $\mathbf{a}_d^r(t_r)$, which is an integer between $[0, 6]$. $\mathbf{a}_d^r(t_r) = 0$ indicates the DC hovers above current unit in the next step, otherwise the DC is assigned to the corresponding neighbor units. Since the flying speeds and transmit powers of DCs are not controlled by GTP, $\mathbf{a}_d^r(t_r)$ contains no action related to DC energy consumption. The $e_d^r(t_r)$ is updated by directly reducing E_δ^r at each GTP step. For the charging process, we assume that the DC keeps charged during the whole charging GTP step and the step period δ_r is long enough for the DC to be fully charged.

2) State Design: According to Section IV-B, the set of average user numbers in all $g \in \mathcal{G}$ at GTP step t_r, $\overline{\mathcal{U}}_g(t_r)$, is used as the state component to represent user distribution changes. This design is realistic since the DA-RAN's central controller, in which the MARL-GTP algorithm is implemented, can collect the data over the whole scenario. For each DC associated with BS b, we use the subset $\overline{\mathcal{U}}_g^b(d, t_r) \subset \overline{\mathcal{U}}_g(t_r)$, which is the set of average user numbers in all $g \in \mathcal{G}_b$, as the state component to reduce complexity due to constraint (11b). The ID of currently served unit $g(d, t_r)$ by the DC is involved in the state to indicate current DC location. The ID of unit where the associated BS is located g_b and the remaining battery energy level $e_d^r(t_r)$ are also involved in the state for DC charging behavior.

By involving all above designs, the state vector for one agent in MARL-GTP algorithm is represented by

$$\mathbf{s}_d^r = \left[\overline{\mathcal{U}}_g^b(d, t_r), g(d, t_r), g_b, e_d^r(t_r)\right]. \tag{13}$$

In case where the dimension of $\overline{\mathcal{U}}_g^b(d, t_r)$ is too large, the "dimension spread" technique is applied to the last three elements in $\mathbf{s}_d^r$ by duplicating them for multiple times, which balances the input weights of different factors.

3) Reward Design: At each step, the summation of $N_k(d, t_r)$ from all DCs is feedback to every DCs as step reward $W^r(t_r)$. On the other hand, to promote the low-power DC flying back to BS for charging, a punishment reward is applied for low-power DC whose value is proportional to the horizontal D2B distance. The general expression of step reward is denoted by

$$W^r(t_r) = \begin{cases} \sum_{d=1}^D \overline{U_d}(t_r), & \frac{e_d^r(t_r)}{E_\delta^r} > |r_{db}(t_r)| \\ \sum_{d=1}^D \overline{U_d}(t_r) - w_{\text{pun}}^r |r_{db}(t_r)|, & \frac{e_d^r(t_r)}{E_\delta^r} \le |r_{db}(t_r)| \\ W_{\text{charge}}^r, & e_d^r(t_r) = |r_{db}(t_r)| = 0 \\ -w_{\text{pun}}^r |r_{db}(t_r)|, & e_d^r(t_r) = 0, |r_{db}(t_r)| \neq 0 \end{cases} \tag{14}$$

where $-w_{\text{pun}}^r$ is the punishment reward constant, $|r_{db}(t_r)|$ is the normalized D2B horizontal distance equals the minimal number of units to be traversed from the DC to the BS. W_{charge}^r is the charging reward granted by reaching the BS for charging when $e_d^r(t_r) = 0$.

4) Priority Experience Replay: The punishment reward in (14) can be regarded as the "reward sharing" [35] technique to promote the agent leaning sparse behaviors with high reward (e.g., W_{charge}^r). To further increase the probability of those high-reward sparse behavior being learned, the priority experience replay (PER) technique [35] is applied into the MARL-GTP algorithm. Instead of randomly choosing experienced state-action pairs in the buffer, the PER stores the past experience in a priority tree, where the state-action pairs with higher step rewards have higher probability to be chosen for learning.

5) Multiagent Fingerprint: In MARL, the convergence is hard to be ensured since the movements of other agents break the environment of each agent into nonstationary [33]. In [33], the nonstationary environment issue in MARL is solved by the multiagent fingerprint technique, which involves the abstracted state information of other agents into each agent's state design.

Apart from the step IDs of other agents suggested by [33], we also involve a neighbor DC indicator vector $\mathbf{o}_d = [o_1, o_2, \ldots, o_6]$ into state design. $\mathbf{o}_d$ contains six elements representing the neighboring DCs' status in the six neighbor units of the DC. Each element in $\mathbf{o}_d$ equals one when its represented unit is occupied by other DC in next GTP step, otherwise the element equals zero. Compared with directly listing all DCs' locations in the state, the $\mathbf{o}_d$ limits the state nonstationary caused by other DCs' movements within six binary dimensions, which highly simplifies the space complexity.

To use $\mathbf{o}_d$, an interagent information exchange process must be executed by all agents in each step, as shown in Fig. 3. In the controller, each agent sequentially takes GTP actions, then the remaining agents keep updating their vector $\mathbf{o}_d$ according to previous agents' actions, and use the latest one in their turns to determine the action. Note that agent 1 in Fig. 3 has

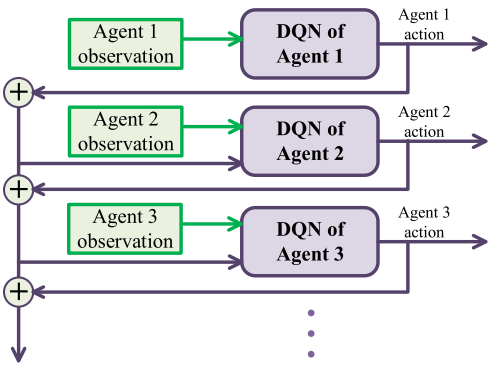


Fig. 3. Interagent information exchange.

no constraint from other DC in its action chosen, while the last agent is constrained by all previous DCs. Such unfairness between state and action spaces of different agents can lead to the "lazy agent problem" [33], i.e., the agent with the most constraints becomes inactive to prevent affecting other agents performance. To overcome the "lazy agent problem," a simple round robin rule is applied to iteratively select the head agent for the interagent information exchange process at each step.

Leveraging the multiagent fingerprint technique, we revise the state design for MARL-GTP as

$$\mathbf{s}_d^r = \left[\overline{\mathcal{U}}_g^b(d, t_r), \mathbf{o}_d, g(d, t_r), g_b, e_d^r(t_r), t_r\right]. \tag{15}$$

In the above equation, we can directly use the current step ID t_r as fingerprint because the MARL-GTP algorithm defines no break behaviors over the episode T, therefore, all DCs can complete each step simultaneously. Algorithm 1 shows the details of the MARL-GTP algorithm within one episode.

Algorithm 1 MARL-GTP Algorithm

Initialize the replay buffer $M^{\mathrm{r}}(d)$ with size $|M^{\mathrm{r}}(d)|$ for each DC.
Initialize the evaluation DQN $Q_d^e(\mathbf{s}_d^{\mathrm{r}}, \mathbf{a}_d^{\mathrm{r}}|\theta_d^e)$ with random parameters θ_d^e for each DC.
Initialize the target DQN $Q_d^t(\mathbf{s}_d^{\mathrm{r}}, \mathbf{a}_d^{\mathrm{r}}|\theta_d^t)$ with random parameters θ_d^t for each DC.
Initialize $\epsilon(d) = 1$, $t_{\mathrm{r}} = 0$
for each episode **do**
  for each t_{r} **do**
    for $d \in \mathcal{D}$ **do**
      Update $\mathbf{o}_d$ with other DCs' actions.
      Choose $\mathbf{a}_d^{\mathrm{r}}(t_{\mathrm{r}})$ from $\mathbf{s}_d^{\mathrm{r}}(t_{\mathrm{r}})$ using ϵ-greedy.
      if take $\mathbf{a}_d^{\mathrm{r}}(t_{\mathrm{r}})$ violating (12a) **then**
        $\mathbf{a}_d^{\mathrm{r}}(t_{\mathrm{r}}) = 0$.
      end if
      Take $\mathbf{a}_d^{\mathrm{r}}(t_{\mathrm{r}})$, obtain reward $W^{\mathrm{r}}(t_{\mathrm{r}})$ according to (14), observe next state $\mathbf{s}_d^{\mathrm{r}}(t_{\mathrm{r}}+1)$.
      Store the transition $\left(\mathbf{s}_d^{\mathrm{r}}(t_{\mathrm{r}}), \mathbf{a}_d^{\mathrm{r}}(t_{\mathrm{r}}), W_d^{\mathrm{r}}(t_{\mathrm{r}}), \mathbf{s}_d^{\mathrm{r}}(t_{\mathrm{r}}+1)\right)$ in $M^{\mathrm{r}}(d)$.
      Sample a training-batch from $M^{\mathrm{r}}(d)$.
      Calculate $\left(\mathbf{s}_d^{\mathrm{r}}(t_{\mathrm{r}}), \mathbf{a}_d^{\mathrm{r}}(t_{\mathrm{r}})\right)$'s target Q value: $y_d^{\mathrm{r}}(t_{\mathrm{r}}) = W_d^{\mathrm{r}}(t_{\mathrm{r}}) + \gamma \max_{\mathbf{a}_d^{\mathrm{r}}} Q_d^t\left(\mathbf{s}_d^{\mathrm{r}}(t+1), \mathbf{a}_d^{\mathrm{r}}|\theta_d^t\right)$.
      Perform gradient decent on θ_d^e to minimize: $\left[y_d^{\mathrm{r}}(t_{\mathrm{r}}) - Q_d^e\left(\mathbf{s}_d^{\mathrm{r}}(t_{\mathrm{r}}), \mathbf{a}_d^{\mathrm{r}}(t_{\mathrm{r}})|\theta_d^e\right)\right]^2$
      if $\mod(t_{\mathrm{r}}, N) = 0$ **then**
        $\theta_d^t = \theta_d^e$.
        $\epsilon(d) = \epsilon(d) - \epsilon_{decay}$
      end if
    end for
    Round robin the order of DCs in inter-agent information exchange.
  end for
end for

C. DEP-LTPRA Algorithm

Given the unit $g_k(d, t_r)$ decided by the higher level MARL-GTP algorithm at GTP step t_r, we propose the DEP-LTPRA algorithm for each DC to serve all users within $g_k(d, t_r)$ over one GTP step period δ_r. To promote the accuracy of LTPRA results, we allow the DC to choose TPRA actions from a continuous action space, which is enabled by the DEP DRL technique. We specify the DEP-LTPRA algorithm with the following designs.

1) Action Design: According to Section III-B, the DC has two types of actions to increase the total achieved throughput, i.e., the transmit power control action, and the trajectory planning action. Considering the high mobility of users, the number of users within the served unit can change with each step t. Since the variable action and state dimensions are not supported by any DRL frameworks, it is hard and inefficient to directly allocate transmit power at the per-user level by the DEP-TPRA algorithm. Therefore, in this work, we define the amount of total transmit power $P_d(t)$ at each step t as the transmit power control action. Given $P_d(t)$, we apply a simple fair power allocation mechanism for all served users in step t, which ensures strict per-user throughput fairness with $m_d(t) = 1$. Specifically, the power allocated to each user u is calculated by

$$P_{du}(t) = \frac{P_d(t)\beta_{du}(t)}{\sum_{u \in \mathcal{U}_d(t)} \beta_{du}(t)}. \tag{16}$$

The choice of $P_d(t)$ is dependent on the current number of users being served, as well as the total available energy within one GTP step period δ_r.

2) State Design: Given the fixed flying height $h_d^{\mathrm{opt}}(t_r)$ defined in Section III-B, the trajectory planning action of the DC is determining its horizontal flying speed $v_d(t)$. We represent $v_d(t)$ by two components $(v_d^x(t), v_d^y(t))$ along x and y-axis in the action space. Define the maximal speed of both components as $V_{\max}^c = V_{\max}/\sqrt{2}$, the values of $v_d^x(t)$ and $v_d^y(t)$ are selected between $[-V_{\max}^c, V_{\max}^c]$. Therefore, the trajectory action space of a DC is defined as a horizontal 2-D square centered at DC's current location with diagonal length $2V_{\max}$. In general, the output action of the DEP-LTPRA algorithm is represented as

$$\mathbf{a}_d^l(t) = \left[v_d^x(t), v_d^y(t), P_d(t)\right]. \tag{17}$$

Considering the variant numbers of users at each step, state $\mathbf{s}_d^l(t)$ has to be designed with a fixed number of dimensions instead of using all users' locations. To find a properer state which indicates the feature of user distribution with a fixed size, the geometric center $c_g(t)$ of all users in the served unit g at step t are selected. We choose $c_g(t)$ as the state component to represent user distribution feature due to the following corollary.

Corollary 1: Given $P_d(t)$ and the per-user power allocation mechanism in (16), the lower bound of DC d's achieved total throughput at step t can be maximized by hovering above the geometric center $c_g(t)$ of all users in $\mathcal{U}_d(t)$.

The detailed proof of Corollary 1 is shown in the Appendix. According to Corollary 1, given different user distributions, flying toward $c_g(t)$ can always be the optimal action to maximize the minimal guaranteed total throughput. Since the purpose of the DEP-TPRA algorithm is to learn the optimal deterministic action for each state, using $c_g(t)$ in state design can not only simplify the state space but also provide guidance information for the DC's trajectory planning at each step. Note that the maximal total throughput at each step t might not be achieved by hovering above $c_g(t)$, $c_g(t)$ only provides a search direction with high probability to find the optimal point nearby. The exact optimal point maximizing total throughput is found by the DEP-LTPRA algorithm. Together with the DC location component $z_d(t)$, as well as the remaining energy of DC within δ_r, the state for the DEP-TPRA algorithm can be represented by

$$\mathbf{s}_d^l(t) = \left[z_d(t), c_g(t), e_d^{\mathrm{l}}(t)\right] \tag{18}$$

where $e_d^{\mathrm{l}}(t) \in [0, E_\delta^r]$.

3) Reward Design: The total throughput achieved by the DC is set as step reward for DEP-TPRA algorithm, together with a negative reward $-w_{\mathrm{pun}}^{\mathrm{l}}$ for states with $E_d^r(t) = 0$

$$W^{\mathrm{l}}(t) = \begin{cases} \sum_{u \in \mathcal{U}_{d(t)}} c_{du}(t), & e_d^l(t) \neq 0 \\ -w_{\mathrm{pun}}^l, & e_d^l(t) = 0. \end{cases} \tag{19}$$

The details of the DEP-TPRA algorithm for one DC are shown in Algorithm 2.

Algorithm 2 DEP-TPRA Algorithm

1: Initialize the replay buffer M^{l} with size $|M^{\mathrm{l}}|$.
2: Initialize the actor network $\Phi_a(\mathbf{s}_d^{\mathrm{l}}|\varphi_a)$ with random parameters φ_a, and target actor network $\Phi_a^t(\mathbf{s}_d^{\mathrm{l}}|\varphi_a^t)$ with parameters $\varphi_a^t = \varphi_a$.
3: Initialize the critic network $\Phi_c(\mathbf{s}_d^{\mathrm{l}}, \mathbf{a}_d^{\mathrm{l}}|\varphi_c)$ with random parameters φ_c, and target critic network $\Phi_c^t(\mathbf{s}_d^{\mathrm{l}}, \mathbf{a}_d^{\mathrm{l}}|\varphi_c^t)$ with parameters $\varphi_c^t = \varphi_c$.
4: **for** each episode **do**
5: **for** each step t **do**
6: Observe current state $\mathbf{s}_d^{\mathrm{l}}(t)$.
7: Choose $\mathbf{a}_d^{\mathrm{l}}(t) = \Phi_a(\mathbf{s}_d^{\mathrm{l}}(t)|\varphi_a) + \mathcal{N}$ where $\mathcal{N}$ is exploration factor.
8: **if** take $\mathbf{a}_d^{\mathrm{l}}(t)$ violating (11b), (11c) **then**
9: Refine the action $\mathbf{a}_d^{\mathrm{l}}(t)$ within the constraints.
10: **end if**
11: Take $\mathbf{a}_d^{\mathrm{l}}(t)$, obtain reward $W^{\mathrm{l}}(t)$ according to (19), observe next state $\mathbf{s}_d^{\mathrm{l}}(t+1)$.
12: Store the transition $\left(\mathbf{s}_d^{\mathrm{l}}(t), \mathbf{a}_d^{\mathrm{l}}(t), W^{\mathrm{l}}(t), \mathbf{s}_d^{\mathrm{l}}(t+1)\right)$ in M^{l}.
13: **end for**
14: Sample a training-batch with N_{bat} transitions from M^{l}.
15: Calculate the critic target by: $y_t(t) = W^{\mathrm{l}}(t) - \gamma\, \Phi_c^t\left(\mathbf{s}_d^{\mathrm{l}}(t+1), \Phi_a^t(\mathbf{s}_d^{\mathrm{l}}(t+1)|\varphi_a^t)\middle|\varphi_c^t\right)$
16: Update critic network parameter φ_c by minimizing critic loss function: $L(\Phi_c) = \frac{1}{N_{\mathrm{bat}}} \sum_t \left[y_t(t) - \varphi_c\left(\mathbf{s}_d^{\mathrm{l}}(t), \mathbf{a}_d^{\mathrm{l}}(t)\middle|\varphi_c\right)\right]^2$.
17: Update actor network parameter φ_a by minimizing actor loss function: $L(\Phi_a) = -\frac{1}{N_{\mathrm{bat}}} \sum_t \varphi_c\left(\mathbf{s}_d^{\mathrm{l}}(t), \Phi_a(\mathbf{s}_d^{\mathrm{l}}(t)|\varphi_a)\middle|\varphi_c\right)$.
18: Soft update target networks: $\varphi_c^t = \tau\varphi_c + (1-\tau)\varphi_c^t$; $\varphi_a^t = \tau\varphi_a + (1-\tau)\varphi_a^t$
19: **end for**

VI. Simulations

To validate the performance of the proposed HDRLTPRA scheme, we build a real-world-based simulation scenario as shown in Fig. 4. The scenario contains all roads in the campus region of University of Waterloo with a size of 2300 m $\times$ 2000 m. Three BSs are included in the scenario, which are represented by solid circles with their communication coverage denoted by dotted lines in Fig. 4. The scenario is divided into 36 units, each hexagon unit is represented by its circumscribed circle. In the simulation, vehicles are considered as high-mobility users. We build the scenario in traffic simulator VISSIM to generate highly authentic vehicle traffic by jointly considering the impacts of traffic signals, driver behaviors, traffic conditions, etc., [36]. The total TPRA task period T is set to six h from 9:00 A.M. to 3:00 P.M. Without loss of generality, we set the GTP step length $\delta_r = 15$ min, and one step t length $\delta_t = 10$ s. The service endurance for each DC T_d is set to 3 h, which is reasonable according to the existing industry products with 5+ endurance time [37]. We compare the performance of our HDRLTPRA scheme with the nonlearning-based baseline scheme in [13]. Detail simulation parameters are shown in Table I.

Figs. 5 and 6 show the trajectory planning results and performance of the MARL-GTP algorithm with two and three DCs, respectively. Each BS is associated by no more than one DC in both scenarios. Figs. 5(a) and 6(a) plot the GTP results of the last three episodes out of 30 000 episodes for both scenarios, in which the global trajectories of different DCs are denoted by dashed lines with different colors. Each number x associated to the units traversed by the trajectories indicates that the DC is planned to serve the unit at the xth GTP step. Since the MARL-GTP algorithm has reached convergence, the trajectory planning results of the last three episodes are identical in both scenarios. Note that the MARL-GTP algorithm tends to reduce overlapping between different DCs' trajectories. In both scenarios, all DCs are assigned to the associated BS units at steps 13 and 25 (marked with black color). Given the 3-h DC service endurance, steps 13 and 25 are exactly the steps at which the DC battery is used up. This effect

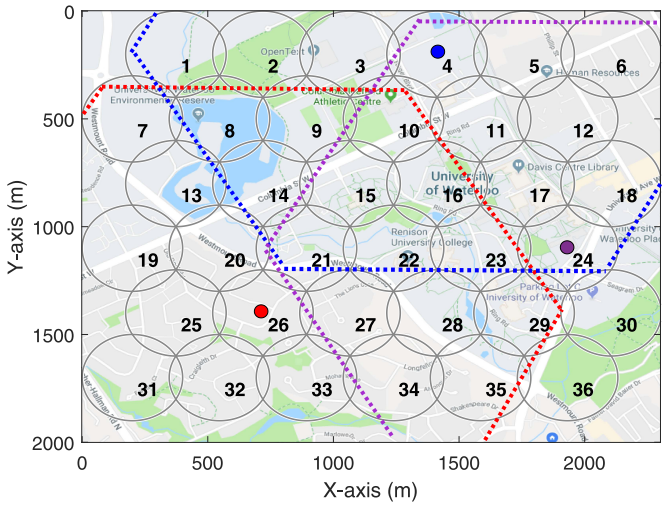


Fig. 4. Real-world-based simulation scenario.

indicates that the charging behavior is successfully learned by the MARL-GTP algorithm.

Figs. 5(b) and 6(b) show the convergence performance for training the MARL-GTP algorithm in both scenarios. We can observe that all DCs' episode rewards reach convergence after 25 000 episodes training. The performance comparisons of the MARL-GTP algorithm and the baseline GTP algorithm in terms of the average number of served users per GTP step are shown in Figs. 5(c) and 6(c), with their mean values denoted by horizontal lines. Note that the MARL-GTP algorithm can serve more users almost at all GTP steps, except for the few steps before the charging step. This effect is caused by the punishment reward design in the MARL-GTP algorithm for learning the charging behavior. In terms of the average number of served users over time period T, around 20%–25% improvements are achieved by the MARL-GTP algorithm when compared with the baseline scheme.

Fig. 7 shows the TPRA results and performance of the DEP-TPRA algorithm in unit 20. The trajectory of the last three episodes over 20 000 episodes training is shown in Fig. 7(a). Note that the three trajectories are different in terms of each step's action choice, but all have the trend to follow the variations of vehicular users' geometric center, as shown in Fig. 7(a). This phenomenon indicates the DEP-TPRA algorithm's capability to tracing the weight center of all users for maximizing total throughput. Fig. 7(b) shows the convergence performance of the DEP-TPRA algorithm. We can observe that the mean value of the episode rewards approximates to 240 after 10 000 episodes training. However, each episode reward still varies between 200 to 250 after their mean value reaching convergence. This indicates that the DEP-LTPRA algorithm can successfully approximate a suboptimal long-term TPRA tread to maximize the accumulative throughput, but still requires longer training time to converge to an exact optimal trajectory. The achieved per-user average throughput at each step (within one episode) of the DEP-TPRA and the baseline algorithms are shown in Fig. 7(c). Since all users in the unit are served by the DC at any step, the trajectory designed by the nonlearning-based baseline algorithm can converge to one point given the statistic vehicle traffic over the whole episode [13]. As shown in Fig. 7(c), the TPRA policy from the DEP-TPRA algorithm achieves higher per-user average throughput than that of the baseline algorithm at most steps, and the DEP-TPRA algorithm overhauls the baseline algorithm by 10% in terms of the achieved mean throughput over all steps.

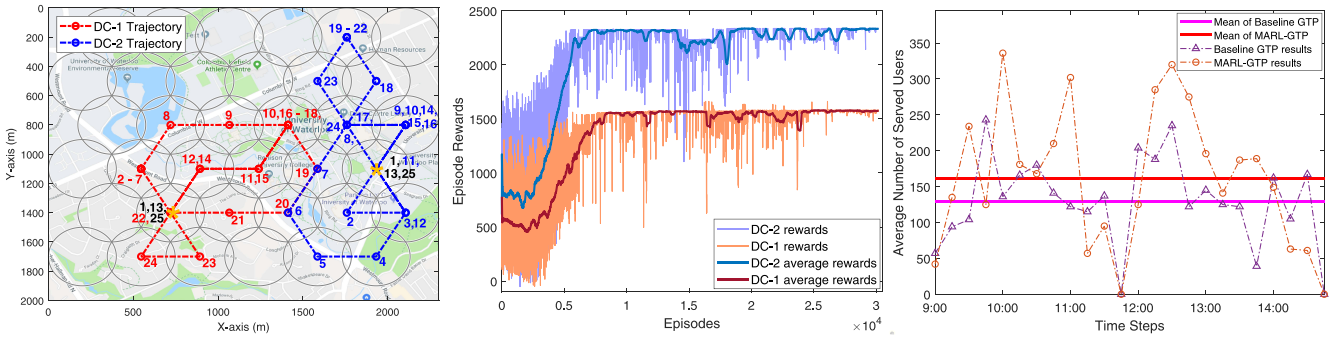
Table II compares the convergence time of proposed algorithms with different numbers of DCs. We can note that the

TABLE I
SIMULATION PARAMETERS

Parameters	Numerical Values
Scenario Parameters	
BS radio coverage radius R_{b}	1200 m
DC radio coverage radius R_{d}	200 m
D2U parameters $(\eta_{\mathrm{LoS}}, \eta_{\mathrm{NLoS}}, a, b)$	(0.1,21,4.88,0.43)
D2B parameters $(\alpha, A, \theta_0, B, \eta_0)$	(3.04,-23.29,-3.61,4.14,20.7)
Carrier frequencies (D2U, D2B)	(2.4 GHz, 850 MHz)
D2B pathloss constraint γ_{DB}	80 dB
DC maximal speed $V_{\max}$	10 m/s
P_d^{prop} parameter $(P_{\mathrm{b}}, P_{\mathrm{i}}, V_{\mathrm{tip}}, V_{\mathrm{h}}, D_0, S_0, \rho, A_0)$	$(577, 793, 200, 7.2, 0.3, 0.05, 1.225, 0.785)$
Full battery Energy E_d	300 Wh
Charging speed p_{crg}	1200 W/s
Noise spectral density σ_0	−174 dBm/Hz
Total bandwidth B_{D}	20 MHz
Maximal per user bandwidth constraint b_{U}	1 MHz
MARL-GTP Parameters	
Number of layers for DQN	3 (except output layer)
Number of nodes for each layer	(128, 64, 64)
Active function	Relu
Minimal ϵ, ϵ decay	(0.001, 0.0002)
Learning rate, reward decay γ	(0.005, 0.9)
Replay buffer size, batch size	(20000, 128)
Maximal episode, steps per episode	(30000, 24)
DDPG-LTPRA Parameters	
Number of layers for actor networks	3
Number of layers for critic networks	2
Number of nodes for actor layers	(30, 30, 3)
Number of nodes for critic layers	(30, 1)
Actor networks active function	(Relu, Relu, tanh)
Critic networks active function	(Relu, Relu)
Learning rates for actor, critic networks	(0.001, 0.002)
Reward decay γ	0.9
Soft update rate τ	0.01
Replay buffer size, batch size	(100000, 512)
Maximal episode, steps per episode	(20000, 90)
Device Parameters	
CPUs	Intel Xeon Gold 6128, 3.4GHz, 4 processors
RAM	128G

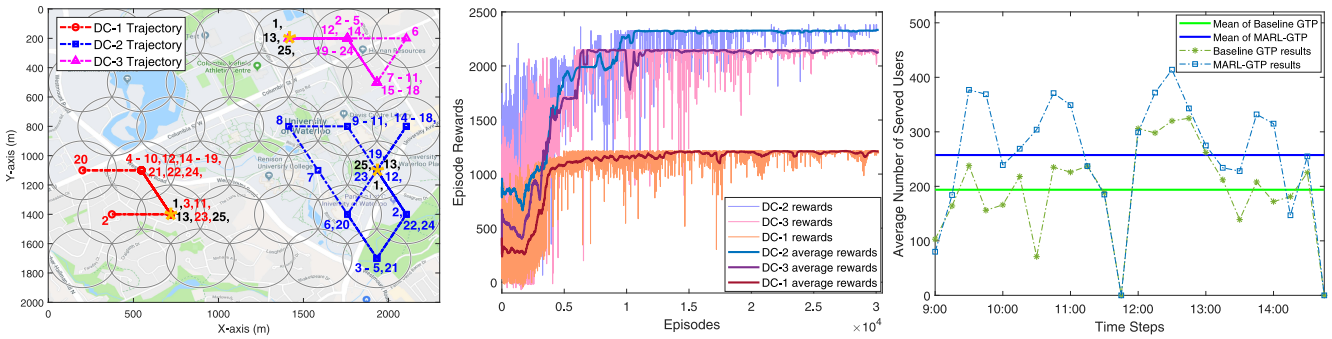
TABLE II
CONVERGENCE PERFORMANCE

Algorithm	DC number	Required episodes	Average convergence time
MARL-GTP	2	25000	30 minutes
	3	23000	30 minutes
	4	21000	28 minutes
DDPG-LTPRA	1	10000	15 minutes



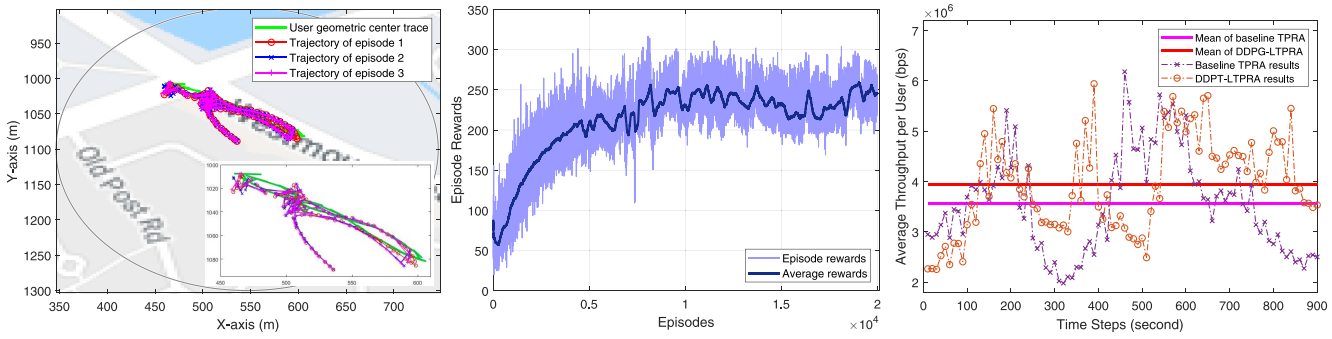
(a) (b) (c)

Fig. 5. GTP of two DCs by MARL-GTP algorithm. (a) Trajectory planning results. (b) Convergence performance. (c) User coverage performance.



(a) (b) (c)

Fig. 6. GTP of three DCs by MARL-GTP algorithm. (a) Trajectory planning results. (b) Convergence performance. (c) User coverage performance.



(a) (b) (c)

Fig. 7. LTPRA of single DC by the DEP-LTPRA algorithm. (a) Trajectory planning results. (b) DEP-LTPRA algorithm convergence performance. (c) Throughput performance.

convergence times of both MARL-GTP algorithm and the DEP-LTPRA algorithm are no more than 30 min given the server parameters listed in Table I. For the MARL-GTP algorithm, the number of episodes required to converge decreases as the number of DCs increases. By increasing the number of drone cells, the number of units associated with each drone cell decreases since more drone cells are involved in sharing the limited unit hexagons. With fewer units to serve, the searching space size of each DC is reduced, thereby simplifying the searching process of the algorithm and leading to less number of episodes required to converge. Although the number of episodes required to converge decreases significantly as drone-cell number increases, the reduction on the real running time of the algorithm is not significant. With more agents interacting with each other in each step, the running time of one episode increases, which compensates the reduced total number of episodes.

By combining the MARL-GTP and DEP-TPRA algorithms, the performance of the HDRLTPRA scheme in terms of total throughput at each step over T is shown in Fig. 8. This simulation is conducted in the two DCs scenario. We can observe that the achieved total throughput is mainly dominated by the number of served users, which is maximized by the MARL-GTP algorithm. Compared with the nonlearning-based solution, the HDRLTPRA scheme can achieve 40% performance promotion in terms of the accumulative network throughput.

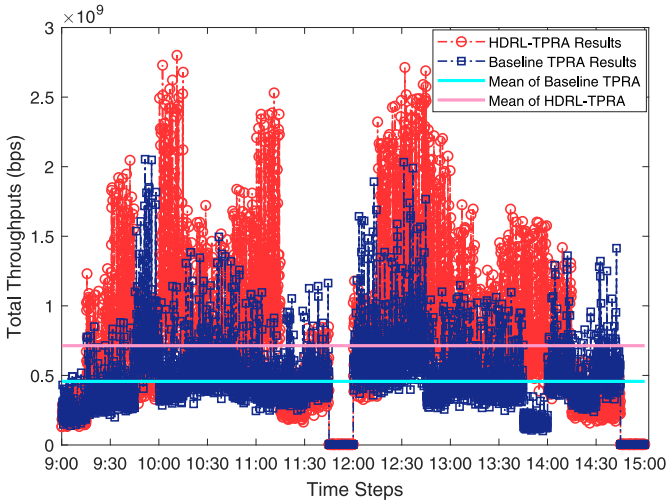


Fig. 8. Accumulated throughput performance of HDRLTPRA scheme.

Note that the hyperparameters of the proposed algorithms should be tuned to suit different scenarios when learning the mobility features of users. For instance, multiple experiments are conducted to determine the 15-min δ_r used in simulations, which can ensure no significant vehicle density vibration in each unit within any t_r. For other highly mobile scenarios with more dynamic user mobility, δ_r should be further reduced to capture the mobility features of the users. Meanwhile, given the same episode length, the number of total time steps increases due to the reduced δ_r. The computing complexity of the MARL-GTP algorithm, therefore, increases since more decisions are required for more time steps.

VII. Conclusion

In this article, we have proposed the HDRLTPRA scheme for multiple DCs to serve high-mobility users over the large area. Based on the hierarchical DRL framework, we have decoupled the multi-DC TPRA problem into two hierarchical subproblems to address the high environment complexity. For the higher level multi-DC GTP subproblem, the MARL-GTP algorithm has been proposed, in which the MARL and multiagent fingerprint techniques are applied to promote convergence in the complex environment. Given the GTP results generated by the MARL-GTP algorithm, the DEP-LTPRA algorithm for lower level LTPRA subproblem has been designed to adapt DC movement and allocate transmit power according to the real-time user distributions. Simulations based on real-world vehicular user traffic show that the HDRLTPRA scheme can enhance the total network throughput by 40% when compared with the nonlearning-based TPRA scheme. The hierarchical DRL-based approach of this work may be used by the DA-RAN operators to deal with highly dynamic and uncertain scenarios. The proposed HDRLTPRA scheme can be implemented to design the TPRA strategies for multiple DCs in highly mobile networks. For future works, fine-grained multitype resource allocation in the HDRLTPRA scheme, and the integration of the HDRLTPRA scheme with satellite-based access networks will be investigated.

Appendix

According to the power allocation mechanism (16) for each user $u \in \mathcal{U}_d(t)$, the achieved total throughput for DC d is

$$C_d(t) = \sum_{u}^{\mathcal{U}_d(t)} b_U \log_2\left[1 + \frac{P_d(t)\beta_{du}(t)}{\sum_{u}^{\mathcal{U}_d(t)} \beta_{du}(t)} \times \frac{\beta_{du}^{-1}(t)}{\sigma_0 b_U}\right] = \sum_{u}^{\mathcal{U}_d(t)} b_U \log_2\left[1 + \frac{P_d(t)}{\sigma_0 b_U \sum_{u\in\mathcal{U}_d(t)} \beta_{du}(t)}\right]. \quad (20)$$

For each step t, given $P_d(t)$, $\sigma_0 b_U$, b_U, $h_d(t) = h_d^{\text{opt}}(t)$ as constants, the maximal $C_d(t)$ is achieved by minimizing $\sum_{u\in\mathcal{U}_d(t)} \beta_{du}(t)$, where

$$\beta_{du}(t) = 10^{\frac{P_{du}^L(r_{du}(t),h_d(t))}{10}} = \left(\frac{4\pi f_c}{c}\right)^2 \left(h_d(t)^2 + r_{du}(t)^2\right) \times 10^{\frac{\text{Pr}_{\text{los}}\eta_{\text{los}} + (1-\text{Pr}_{\text{los}})\eta_{\text{nlos}}}{10}}. \quad (21)$$

According to (2), it is easy to prove that the component $\beta_{\text{los}} = 10^{[(\text{Pr}_{\text{los}}\eta_{\text{los}} + (1-\text{Pr}_{\text{los}})\eta_{\text{nlos}})/10]}$ is a monotonic increasing function of D2U horizontal distance $r_{du}(t)$. Given the system model, the maximal D2U horizontal distance can be the diameter of one unit's circumscribed circle $2 \times R_d$, which maximizes the $\max(\beta_{\text{los}}) = \beta_{\text{los}}^{\max}$. By applying the maximal β_{los} to all users' path-loss calculation, the upper bound of all users' path-loss summation at step t is calculated by

$$UB\left(\sum_{u}^{\mathcal{U}_d(t)} \beta_{du}(t)\right) = C \sum_{u}^{\mathcal{U}_d(t)} \left(h_d^{\text{opt}}(t)^2 + r_{du}(t)^2\right) = C \sum_{u}^{\mathcal{U}_d(t)} r_{du}(t)^2 + C|\mathcal{U}_d(t)|h_d^{\text{opt}}(t)^2 \quad (22)$$

where $C = \beta_{\text{los}}^{\max}(16\pi^2 f_c^2/c^2)$. Since the second component in (22) $C|\mathcal{U}_d(t)|h_d^{\text{opt}}(t)^2$ is constant, the minimal upper bound can be calculated by minimizing $\sum_{u}^{\mathcal{U}_d(t)} r_{du}(t)^2$, which is achieved by the geometric center of all users. Note that the achieved total throughput $C_d(t)$ is lower bounded by using the upper bound defined in (22), therefore, we have proven that the lower bound of achieved total throughput $C_d(t)$ of DC d at step t can be maximized when the DC hovering above $c_g(t)$.

References

[1] M. Mozaffari, A. T. Z. Kasgari, W. Saad, M. Bennis, and M. Debbah, "Beyond 5G with UAVs: Foundations of a 3D wireless cellular network," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 357–372, Jan. 2019.

[2] X. Shen *et al.*, "AI-assisted network-slicing based next-generation wireless networks," *IEEE Open J. Veh. Technol.*, vol. 1, no. 1, pp. 45–66, Jan. 2020.

[3] Z. Xiong, Y. Zhang, D. Niyato, R. Deng, P. Wang, and L. Wang, "Deep reinforcement learning for mobile 5G and beyond: Fundamentals, applications, and challenges," *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 44–52, Jun. 2019.

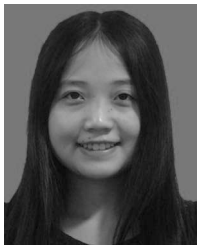
[4] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, Apr. 2019.

[5] H. Wu, Z. Wei, Y. Hou, N. Zhang, and X. Tao, "Cell-edge user offloading via flying UAV in non-uniform heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2411–2426, Apr. 2020.

[6] M. Chen, W. Saad, and C. Yin, "Echo-liquid state deep learning for 360○ content transmission and caching in wireless VR networks with cellular-connected UAVs," *IEEE Trans. Commun.*, vol. 67, no. 9, pp. 6386–6400, Sep. 2019.
[7] H. Wu, F. Lyu, C. Zhou, J. Chen, L. Wang, and X. Shen, "Optimal UAV caching and trajectory in aerial-assisted vehicular networks: A learning-based approach," *IEEE J. Sel. Areas Commun.*, early access, Jun. 29, 2020, doi: 10.1109/JSAC.2020.3005469.
[8] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2714–2741, 4th Quart., 2018.
[9] A. Dhekne, M. Gowda, and R. R. Choudhury, "Cell tower extension through drones: Poster," in *Proc. ACM MobiCom*, 2016, pp. 456–457.
[10] "Study on enhanced LTE support for aerial vehicles, V15.0.0," 3GPP, Sophia Antipolis, France, Rep. TR 36.777, 2018.
[11] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.
[12] S. Zhang, H. Zhang, B. Di, and L. Song, "Cellular UAV-to-X communications: Design and optimization for multi-UAV networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1346–1359, Feb. 2019.
[13] W. Shi *et al.*, "Multi-drone 3D trajectory planning and scheduling in drone-assisted radio access networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8145–8158, Aug. 2019.
[14] N. Kato *et al.*, "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 140–147, Aug. 2019.
[15] S. Yin, S. Zhao, Y. Zhao, and F. R. Yu, "Intelligent trajectory design in UAV-aided communications with reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8227–8231, Aug. 2019.
[16] U. Challita, W. Saad, and C. Bettstetter, "Cellular-connected UAVs over 5G: Deep reinforcement learning for interference management," 2018. [Online]. Available: arXiv:1801.05500.
[17] X. Lu, L. Xiao, and C. Dai, "UAV-aided 5G communications with deep reinforcement learning against jamming," 2019. [Online]. Available: arXiv:1805.06628.
[18] J. Hu, H. Zhang, and L. Song, "Reinforcement learning for decentralized trajectory design in cellular UAV networks with sense-and-send protocol," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6177–6189, Aug. 2019.
[19] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.
[20] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient Internet of Things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, Nov. 2017.
[21] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang, and X. Shen, "Software defined space-air-ground integrated vehicular networks: Challenges and solutions," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 101–109, Jul. 2017.
[22] Q. Zhang, M. Jiang, Z. Feng, W. Li, W. Zhang, and M. Pan, "IoT enabled UAV: Network architecture and routing algorithm," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3727–3742, Apr. 2019.
[23] L. Song, D. Niyato, Z. Han, and E. Hossain, "Game-theoretic resource allocation methods for device-to-device communication," *IEEE Wireless Commun.*, vol. 21, no. 3, pp. 136–144, Jun. 2014.
[24] R. Zhang, L. Song, Z. Han, and B. Jiao, "Physical layer security for two-way untrusted relaying with friendly jammers," *IEEE Trans. Veh. Technol.*, vol. 61, no. 8, pp. 3693–3704, Oct. 2012.
[25] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, and X. Shen, "Energy-efficient UAV-assisted mobile edge computing: Resource allocation and trajectory optimization," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3424–3438, Mar. 2020.
[26] H. Wang, H. Zhao, W. Wu, J. Xiong, D. Ma, and J. Wei, "Deployment algorithms of flying base stations: 5G and beyond with UAVs," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10009–10027, Dec. 2019.
[27] N. Cheng *et al.*, "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.
[28] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal lap altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.
[29] A. Al-Hourani and K. Gomez, "Modeling cellular-to-UAV path-loss for suburban environments," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 82–85, Feb. 2018.
[30] R. K. Jain, D. W. Chiu, and W. R. Hawe, *A Quantitative Measure of Fairness and Discrimination*, Eastern Res. Lab., Digit. Equipment Corp., Hudson, MA, USA, 1984.
[31] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.
[32] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," in *Proc. NeurIPS*, 2016, pp. 3675–3683.
[33] J. Foerster *et al.*, "Stabilising experience replay for deep multi-agent reinforcement learning," in *Proc. ICML*, vol. 70, 2017, pp. 1146–1155.
[34] F. Wu, H. Zhang, J. Wu, and L. Song, "Cellular UAV-to-device communications: Trajectory design and mode selection by multi-agent deep reinforcement learning," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4175–4189, Jul. 2020.
[35] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proc. ICML*, vol. 99, pp. 278–287, Jun. 1999.
[36] N. Cheng *et al.*, "A comprehensive simulation platform for space-air-ground integrated network," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 178–185, Feb. 2020.
[37] *Perimeter 8*. Accessed: Apr. 5, 2020. [Online]. Available: https://skyfront.com/perimeter-8

Weisen Shi (Graduate Student Member, IEEE) received the B.S. degree from Tianjin University, Tianjin, China, in 2013, the M.S. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2016, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2020.

His interests include space-air-ground integrated networks, UAV communication and networking, and RAN slicing.



Junling Li (Graduate Student Member, IEEE) received the B.S. degree from Tianjin University, Tianjin, China, in 2013, the M.S. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2016, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2020.

She is currently a Joint Postdoctoral Research Fellow with Shenzhen Institute of Artificial Intelligence and Robotics for Society, Chinese University of Hong Kong, Shenzhen, China, and the University of Waterloo. Her interests include game theory, machine learning, software-defined networking, network function virtualization, and vehicular networks.

Dr. Li received the Best Paper Award at the IEEE/CIC International Conference on Communications in China in 2019.

Huaqing Wu (Graduate Student Member, IEEE) received the B.E. and M.E. degrees from Beijing University of Posts and Telecommunications, Beijing, China, in 2014 and 2017, respectively. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

Her current research interests include vehicular networks with emphasis on edge caching, resource allocation, and space-air-ground integrated networks.

Conghao Zhou (Graduate Student Member, IEEE) received the B.S. degree from Northeastern University, Shenyang, China, in 2017, and the M.S. degree from the University of Illinois at Chicago, Chicago, IL, USA, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

His research interests include space-air-ground integration networks and machine learning in wireless networks.

Nan Cheng (Member, IEEE) received the B.E. and M.S. degrees from the Department of Electronics and Information Engineering, Tongji University, Shanghai, China, in 2009 and 2012, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2016.

He is currently a Professor with the School of Telecommunication Engineering, Xidian University, Xi'an, China. He worked as a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, and the Department of Electrical and Computer Engineering, University of Waterloo from 2017 to 2018. His current research focuses on space-air-ground integrated system, big data in vehicular networks, and self-driving system. His research interests also include performance analysis, MAC, opportunistic communication, and application of AI for vehicular networks.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular *ad hoc* and sensor networks.

Prof. Shen received the R. A. Fessenden Award from IEEE, Canada, in 2019, the Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) presented in 2019, the James Evans Avant Garde Award from the IEEE Vehicular Technology Society in 2018, the Joseph LoCicero Award in 2015, the Education Award from the IEEE Communications Society in 2017, the Technical Recognition Award from Wireless Communications Technical Committee in 2019, the AHSN Technical Committee in 2013, the Excellent Graduate Supervision Award from the University of Waterloo in 2006, and the Premier's Research Excellence Award from the Province of Ontario, Canada, in 2003. He was/is the Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, *IET Communications*, and *Peer-to-Peer Networking and Applications*. He served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, and IEEE Globecom'07, the Symposia Chair for the IEEE ICC'10, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He is the Elected IEEE Communications Society Vice President for Technical and Educational Activities, the Vice President for Publications, the Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a Member of IEEE Fellow Selection Committee. He is a Registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.