Deep Reinforcement Learning for Interference-Aware Path Planning of Cellular-Connected UAVs

Ursula Challita*, Walid Saad†, and Christian Bettstetter‡

*School of Informatics, The University of Edinburgh, Edinburgh, UK. Email: ursula.challita@ed.ac.uk.

†Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA. Email: walids@vt.edu.

‡Alpen-Adria-Universität Klagenfurt, Institute of Networked and Embedded Systems, Klagenfurt, Austria. Email: christian.bettstetter@aau.at.

***Abstract*—Providing wireless cellular connectivity for unmanned aerial vehicle (UAV) user equipments (UEs) is contingent upon proper management of their resulting interference. To this end, in this paper, an interference-aware path planning scheme for a multi-UAV network is proposed. In particular, each UAV aims at achieving a tradeoff between minimizing energy efficiency, wireless latency, and the interference level caused on the ground network along its path. The problem is cast as a dynamic game among UAVs. To solve this game, a deep reinforcement learning algorithm, based on echo state network (ESN) cells, is proposed. The suggested deep ESN architecture is trained to allow each UAV to map each observation of the network state to an action, with the goal of minimizing a sequence of time-dependent utility functions. Each UAV uses ESN to learn its optimal path, transmission power level, and cell association vector at different locations along its path. The suggested algorithm is shown to reach a subgame perfect Nash equilibrium upon convergence. Simulation results show that the proposed scheme achieves better wireless latency per UAV and rate per ground UE while requiring a number of steps that is comparable to a heuristic baseline that considers moving via the shortest distance towards the corresponding destinations.**

I. Introduction

Cellular-connected unmanned aerial vehicles (UAVs) will be an integral component of future's wireless networks as evidenced by recent interest from academia, industry, and 3GPP standardizations [1]–[6]. Such UAV-user equipments (UEs) will enable a myriad of applications ranging from real-time video streaming to surveillance. Nevertheless, the ability of UAV-UEs to establish line-of-sight (LoS) connectivity to cellular base stations (BSs) is both a blessing and a curse. On the one hand, it enables high-speed data access. On the other hand, it can lead to substantial inter-cell mutual interference among the UAVs and to the ground users. As such, a wide-scale deployment of UAV-UEs is only possible if interference management challenges are addressed [3]–[5].

While some literature has recently studied the use of UAVs as mobile BSs [7]–[10], analysis of UAV-UEs (*short-handed hereinafter as UAVs*) remains relatively scarce [3]–[5]. For instance, in [3], the authors study the impact of UAVs on the uplink performance of a ground LTE network. In [4], the authors use measurements and ray tracing simulations to study the airborne connectivity requirements and propagation characteristics of UAV-UEs. The authors in [5] analyze the coverage probability of the downlink of a cellular network that serves both aerial and ground users. Nevertheless, this prior art is limited to studying the impact that cellular connected UAVs have on the ground network. Indeed, the existing prior art does not provide a solution for optimizing the performance of a cellular network that serves both aerial and ground UEs in order to overcome the interference challenge that arises in this context. UAV trajectory optimization is essential in such scenarios. It allows UAVs to adapt their movement paths based the rate requirements of the aerial and ground UEs, thus improving the overall network performance. The problem of UAV path planning has been studied earlier, however, for non-UAV-UE applications [8]–[11]. In [8], the authors propose a distributed path planning algorithm for multiple UAVs to deliver delay-sensitive information to different ad-hoc nodes. The authors in [9] optimize a UAV's trajectory in an energy-efficient manner. The authors in [10] propose a mobility model that combines area coverage, network connectivity, and UAV energy constraints for path planning. In [11], the authors propose a fog-networking-based system architecture to coordinate a network of UAVs for video services in sports events. However, despite being interesting, this body of work [8]–[11] is restricted to UAVs as BSs and does not account for UAV-UEs and their associated interference challenges. Hence, the approaches proposed therein cannot readily be used for cellular connected UAVs.

The main contribution of this paper is a novel deep reinforcement learning (RL) framework based on echo state network (ESN) cells for optimizing the trajectories of multiple cellular-connected UAVs in an online manner. This framework will allow the UAVs to minimize the interference they cause on the ground network as well as their wireless transmission latency. To realize this, we propose a noncooperative game in which the players are the UAVs and the objective of each UAV is to *autonomously* and *jointly* learn its path, transmit power level, and association vector. A major challenge in this game is the need for each UAV to have full knowledge of the ground network topology, ground UEs service requirements, and other UAVs' locations. Therefore, to solve this game, we propose a deep RL ESN-based algorithm, using which the UAVs can predict the dynamics of the network and subsequently determine their optimal paths as well as the allocation of their resources along their paths. In essence, two important features of our proposed algorithm are *adaptation* and *generalization*. Indeed, UAVs can take decisions for *unseen* network states, based on the reward they got from previous states. This is mainly due to the use of ESN cells which enable the UAVs to retain their previous memory states. To our best knowledge, *this is the first work that exploits the framework of deep ESN for interference-aware path planning of cellular-connected UAVs*. Simulation results show that the proposed approach improves the tradeoff between energy efficiency, wireless latency, and the interference level caused on the ground network.

The rest of this paper is organized as follows. Section II presents the system model. Section III describes the proposed noncooperative game model. The deep RL ESN-based algorithm is proposed in Section IV. In Section V, simulation results are analyzed. Finally, conclusions are drawn in Section IV.

II. System Model

Consider the uplink (UL) of a wireless cellular network composed of a set $\mathcal{S}$ of S ground BSs, a set $\mathcal{Q}$ of Q ground UEs, and a set $\mathcal{J}$ of J cellular-connected UAVs. The UL is defined as the link from UE q or UAV j to BS s. Each BS

$s \in \mathcal{S}$ serves a set $\mathcal{K}_s \subseteq \mathcal{Q}$ of K_s UEs and a set $\mathcal{N}_s \subseteq \mathcal{J}$ of N_s cellular-connected UAVs. The total system bandwidth, B, is divided into a set $\mathcal{C}$ of C resource blocks (RBs). Each UAV $j \in \mathcal{N}_s$ is allocated a set $\mathcal{C}_{j,s} \subseteq \mathcal{C}$ of $C_{j,s}$ RBs and each UE $q \in \mathcal{K}_s$ is allocated a set $\mathcal{C}_{q,s} \subseteq \mathcal{C}$ of $C_{q,s}$ RBs by its serving BS s. At each BS s, a particular RB $c \in \mathcal{C}$ is allocated to *at most* one UAV $j \in \mathcal{N}_s$, or UE $q \in \mathcal{K}_s$.

An airborne Internet of Things (IoT) is considered in which the UAVs are equipped with different IoT devices, such as cameras, sensors, and GPS that can be used for various applications such as surveillance, monitoring, delivery and real-time video streaming. The 3D coordinates of each UAV $j \in \mathcal{J}$ and each ground user $q \in \mathcal{Q}$ are, respectively, (x_j, y_j, z_j) and $(x_q, y_q, 0)$. All UAVs are assumed to fly at a fixed altitude z_j above the ground (as done in [9], [12], [13]) while the horizonal coordinates (x_j, y_j) of each UAV j vary in time. Each UAV j needs to move from an initial location o_j to a final destination d_j while transmitting *online* its mission-related data such as sensor recordings, video streams, and location updates. We assume that the initial and final locations of each UAV are pre-determined based on its mission objectives.

For ease of exposition, we consider a virtual grid for the mobility of the UAVs. We discretize the space into a set $\mathcal{A}$ of A equally sized unit areas. The UAVs move along the center of the areas $c_a = (x_a, y_a, z_a)$, which yields a finite set of possible paths $\boldsymbol{p}_j$ for each UAV j. The path $\boldsymbol{p}_j$ of each UAV j is defined as a sequence of area units $\boldsymbol{p}_j = (a_1, a_2, \cdots, a_l)$ such that $a_1 = o_j$ and $a_l = d_j$. The area size of the discretized area units $(a_1, a_2, \cdots, a_A) \in \mathcal{A}$ is chosen to be sufficiently small such that the UAVs' locations can be assumed to be approximately constant within each area even at the maximum UAV's speed, as commonly done in the literature [12]. We assume a constant speed $0 < V_j \leq \widehat{V}_j$ for each UAV where $\widehat{V}_j$ is the maximum speed of UAV j. Therefore, the time required by each UAV to travel between any two unit areas is constant.

A. Channel Models

We consider the sub-6 GHz band and the free-space path loss model for the UAV-BS data link. The path loss between UAV j at location a and BS s, $\xi_{j,s,a}$, is given by [14]:

$$\xi_{j,s,a}(\text{dB}) = 20\ \log_{10}(d_{j,s,a}) + 20\ \log_{10}(f_c) - 147.55, \quad (1)$$

where f_c is the system center frequency and $d_{j,s,a}$ is the Euclidean distance between UAV j at location a and BS s. We consider a Rician distribution for modeling the small-scale fading between UAV j and ground BS s thus accounting for the LoS and multipath scatterers that can be experienced at the BS. In particular, adopting the Rician channel model for the UAV-BS link is validated by the fact that the channel between a given UAV and a ground BS is mainly dominated by a LoS link [9]. We assume that the Doppler spread due to the mobility of the UAVs is compensated for based on existing techniques such as frequency synchronization using a phase-locked loop [15] as done in [12] and [9].

For the terrestrial UE-BS links, we consider a Rayleigh fading channel. For a carrier frequency, f_c, of 2 GHz, the path loss between UE q and BS s is given by [16]:

$$\zeta_{q,s}(\text{dB}) = 15.3 + 37.6\ \log_{10}(d_{q,s}), \quad (2)$$

where $d_{q,s}$ is the Euclidean distance between UE q and BS s.

The average signal-to-interference-plus-noise ratio (SINR), $\Gamma_{j,s,c,a}$, of the UAV-BS link between UAV j at location a ($a \in \mathcal{A}$) and BS s over RB c will be:

$$\Gamma_{j,s,c,a} = \frac{P_{j,s,c,a} h_{j,s,c,a}}{I_{j,s,c} + B_c N_0}, \quad (3)$$

where $P_{j,s,c,a} = \widehat{P}_{j,s,a}/C_{j,s}$ is the transmit power of UAV j at location a to BS s over RB c and $\widehat{P}_{j,s,a}$ is the total transmit power of UAV j to BS s at location a. Here, the total transmit power of UAV j is assumed to be distributed uniformly among all of its associated RBs. $h_{j,s,c,a} = g_{j,s,c,a} 10^{-\xi_{j,s,a}/10}$ is the channel gain between UAV j and BS s on RB c at location a where $g_{j,s,c,a}$ is the Rician fading parameter. N_0 is the noise power spectral density and B_c is the bandwidth of an RB c. $I_{j,s,c} = \sum_{r=1,r\neq s}^{S}(\sum_{k=1}^{K_r} P_{k,r,c}h_{k,s,c} + \sum_{n=1}^{N_r} P_{n,r,c,a'}h_{n,s,c,a'})$ is the total interference power on UAV j at BS s when transmitting over RB c, where $\sum_{r=1,r\neq s}^{S}\sum_{k=1}^{K_r} P_{k,r,c}h_{k,s,c}$ and $\sum_{r=1,r\neq s}^{S}\sum_{n=1}^{N_r} P_{n,r,c,a'}h_{n,s,c,a'}$ correspond, respectively, to the interference from the K_r UEs and the N_r UAVs (at their respective locations a') connected to neighboring BSs r and transmitting using the same RB c as UAV j. $h_{k,s,c} = m_{k,s,c} 10^{-\zeta_{k,s}/10}$ is the channel gain between UE k and BS s on RB c where $m_{k,s,c}$ is the Rayleigh fading parameter. Therefore, the achievable data rate of UAV j at location a associated with BS s can be defined as $R_{j,s,a} = \sum_{c=1}^{C_{j,s}} B_c \log_2(1 + \Gamma_{j,s,c,a})$.

Given the achievable data rate of UAV j and assuming that each UAV is an M/D/1 queueing system, the corresponding latency over the UAV-BS wireless link is given by [17]:

$$\tau_{j,s,a} = \frac{\lambda_{j,s}}{2\mu_{j,s,a}(\mu_{j,s,a} - \lambda_{j,s})} + \frac{1}{\mu_{j,s,a}}, \quad (4)$$

where $\lambda_{j,s}$ is the average packet arrival rate (packets/s) traversing link (j, s) and originating from UAV j. $\mu_{j,s,a} = R_{j,s,a}/\nu$ is the service rate over link (j, s) at location a where ν is the packet size. On the other hand, the achievable data rate for a ground UE q served by BS s is given by:

$$R_{q,s} = \sum_{c=1}^{C_{q,s}} B_c \log_2\Big(1 + \frac{P_{q,s,c} h_{q,s,c}}{I_{q,s,c} + B_c N_0}\Big), \quad (5)$$

where $h_{q,s,c} = m_{q,s,c} 10^{-\zeta_{q,s}/10}$ is the channel gain between UE q and BS s on RB c and $m_{q,s,c}$ is the Rayleigh fading parameter. $P_{q,s,c} = \widehat{P}_{q,s}/C_{q,s}$ is the transmit power of UE q to its serving BS s on RB c and $\widehat{P}_{q,s}$ is the total transmit power of UE q. Here, we also consider equal power allocation among the allocated RBs for the ground UEs. $I_{q,s,c} = \sum_{r=1,r\neq s}^{S}(\sum_{k=1}^{K_r} P_{k,r,c}h_{k,s,c} + \sum_{n=1}^{N_r} P_{n,r,c,a'}h_{n,s,c,a'})$ is the total interference power experienced by UE q at BS s on RB c where $\sum_{r=1,r\neq s}^{S}\sum_{k=1}^{K_r} P_{k,r,c}h_{k,s,c}$ and $\sum_{r=1,r\neq s}^{S}\sum_{n=1}^{N_r} P_{n,r,c,a'}h_{n,s,c,a'}$ correspond, respectively, to the interference from the K_r UEs and the N_r UAVs (at their respective locations a') associated with the neighboring BSs r and transmitting using the same RB c as UE q.

B. Problem Formulation

Our objective is to find the optimal path for each UAV j based on its mission objectives as well as its interference on the ground network. Thus, we seek to minimize: a) the interference level that each UAV causes on the ground UEs and other UAVs, b) the transmission delay over the wireless link, and c) the time needed to reach the destination. To realize this, we optimize the paths of the UAVs jointly with the cell association vector and power control at each location $a \in \mathcal{A}$ along each UAV's path. We consider a directed graph $G_j = (\mathcal{V}, \mathcal{E}_j)$ for each UAV j where $\mathcal{V}$ is the set of vertices corresponding to the centers of the unit areas $a \in \mathcal{A}$ and $\mathcal{E}_j$ is the set of edges formed along the path of UAV j. We let $\widehat{\boldsymbol{P}}$ be the transmission power vector with each element $\widehat{P}_{j,s,a} \in [0, \overline{P}_j]$ being the transmission power level of UAV j to its serving BS s at location a where $\overline{P}_j$ is the maximum transmission power of UAV j. $\boldsymbol{\alpha}$ is the path formation vector with each element $\alpha_{j,a,b} \in \{0,1\}$ indicating whether or not a directed link is formed from area a towards area b for UAV j, i.e., if UAV j moves from a to b along its path. $\boldsymbol{\beta}$ is the UAV-BS association vector with each element $\beta_{j,s,a} \in \{0,1\}$ denoting whether or not UAV j is associated with BS s at location a. Next, we present our optimization problem whose goal is to determine the path of each UAV along with its cell association vector and its transmit power level at each location a along its path $\boldsymbol{p}_j$:

$$\min_{\widehat{\boldsymbol{P}},\boldsymbol{\alpha},\boldsymbol{\beta}} \varrho \sum_{j=1}^{J}\sum_{s=1}^{S}\sum_{c=1}^{C_{j,s}}\sum_{a=1}^{A}\sum_{r=1,r\neq s}^{S} \frac{\widehat{P}_{j,s,a} h_{j,r,c,a}}{C_{j,s}}$$

$$+ \varpi \sum_{j=1}^{J}\sum_{a=1}^{A}\sum_{b=1,b\neq a}^{A} \alpha_{j,a,b} + \phi \sum_{j=1}^{J}\sum_{s=1}^{S}\sum_{a=1}^{A} \beta_{j,s,a}\tau_{j,s,a}, \quad (6)$$

$$\sum_{b=1,b\neq a}^{A} \alpha_{j,b,a} \leq 1 \;\; \forall j \in \mathcal{J}, a \in \mathcal{A}, \quad (7)$$

$$\sum_{a=1,a\neq o_j}^{A} \alpha_{j,o_j,a}{=}1 \;\; \forall j \in \mathcal{J}, \sum_{a=1,a\neq d_j}^{A} \alpha_{j,a,d_j}{=}1 \;\; \forall j \in \mathcal{J}, \quad (8)$$

$$\sum_{a=1,a\neq b}^{A} \alpha_{j,a,b} - \sum_{f=1,f\neq b}^{A} \alpha_{j,b,f}{=}0 \; \forall j \in \mathcal{J}, b \in \mathcal{A} \; (b \neq o_j, b \neq d_j), \quad (9)$$

$$\widehat{P}_{j,s,a} \geq \sum_{b=1,b\neq a}^{A} \alpha_{j,b,a} \;\; \forall j \in \mathcal{J}, s \in \mathcal{S}, a \in \mathcal{A}, \quad (10)$$

$$\widehat{P}_{j,s,a} \geq \beta_{j,s,a} \;\; \forall j \in \mathcal{J}, s \in \mathcal{S}, a \in \mathcal{A}, \quad (11)$$

$$\sum_{s=1}^{S} \beta_{j,s,a} - \sum_{b=1,b\neq a}^{A} \alpha_{j,b,a} = 0 \;\; \forall j \in \mathcal{J}, a \in A, \quad (12)$$

$$\sum_{c=1}^{C_{j,s}} \Gamma_{j,s,c,a} \geq \beta_{j,s,a}\overline{\Gamma}_j \;\; \forall j \in \mathcal{J}, s \in \mathcal{S}, a \in \mathcal{A}, \quad (13)$$

$$0 \leq \widehat{P}_{j,s,a} \leq \overline{P}_j \;\; \forall j \in \mathcal{J}, s \in \mathcal{S}, a \in \mathcal{A}, \quad (14)$$

$$\alpha_{j,a,b} \in \{0,1\}, \beta_{j,s,a} \in \{0,1\} \;\; \forall j \in \mathcal{J}, s \in \mathcal{S}, \; a,b \in \mathcal{A}. \quad (15)$$

The objective function in (6) captures the total interference level that the UAVs cause on neighboring BSs along their paths, the length of the paths of the UAVs, and their wireless transmission delay. ϱ, ϖ and ϕ are multi-objective weights used to control the tradeoff between the three considered metrics. These weights can be adjusted to meet the requirements of each UAV's mission. For instance, the time to reach the destination is critical in search and rescue applications while the latency is important for online video streaming applications. (7) guarantees that each area a is visited by UAV j at most once along its path $\boldsymbol{p}_j$. (8) guarantees that the trajectory of each UAV j starts at its initial location o_j and ends at its final destination d_j. (9) guarantees that if UAV j visits area b, it should also leave from area b ($b \neq o_j, b \neq d_j$). (10) and (11) guarantee that UAV j transmits to BS s at area a with power $\widehat{P}_{j,s,a} > 0$ only if UAV j visits area a, i.e., $a \in \boldsymbol{p}_j$ and such that j is associated with BS s at location a. (12) guarantees that each UAV j is associated with one BS s at each location a along its path $\boldsymbol{p}_j$. (13) guarantees an upper limit, $\overline{\Gamma}_j$, for the SINR value $\Gamma_{j,s,c,a}$ of the transmission link from UAV j to BS s on RB c at each location a, $a \in \boldsymbol{p}_j$. This, in turn, ensures successful decoding of the transmitted packets at the serving BS. The value of $\overline{\Gamma}_j$ is application and mission specific. We note that the SINR check at each location a is valid for our problem due to the consideration of small-sized area units. (14) and (15) represent the feasibility constraints. The formulated optimization problem is a mixed integer non-linear program, which is computationally complex to solve for large networks.

To address this challenge, we adopt a distributed approach in which each UAV decides autonomously on its next path location along with its corresponding transmit power and association vector. In fact, a centralized approach requires control signals to be transmitted to the UAVs at all time. This might incur high round-trip latencies that are not desirable for real-time applications such as online video streaming. Further, a centralized approach requires a central entity to have full knowledge of the current state of the network and the ability to communicate with all UAVs at all time. However, this might not be feasible in case the UAVs belong to different operators or in scenarios in which the environment changes dynamically. Therefore, we next propose a distributed approach for each UAV j to learn its path $\boldsymbol{p}_j$ along with its transmission power level and association vector at each location a along its path in an autonomous and online manner.

III. Towards a Self-Organizing Network of an Airborne Internet of Things

A. Game-Theoretic Formulation

Our objective is to develop a distributed approach that allows each UAV to take actions in an autonomous and online manner. For this purpose, we model the multi-agent path planning problem as a finite dynamic noncooperative game model $\mathcal{G}$ with perfect information [18]. Formally, we define the game as $\mathcal{G} = (\mathcal{J}, \mathcal{T}, \mathcal{Z}_j, \mathcal{V}_j, \Pi_j, u_j)$ with the set $\mathcal{J}$ of UAVs being the agents. $\mathcal{T}$ is a finite set of stages which correspond to the steps required for all UAVs to reach their sought destinations. $\mathcal{Z}_j$ is the set of actions that can be taken by UAV j at each $t \in \mathcal{T}$, $\mathcal{V}_j$ is the set of all observed network states by UAV j up to stage T, Π_j is a set of probability distributions defined over all $z_j \in \mathcal{Z}_j$, and u_j is the payoff function of UAV j. At each stage $t \in \mathcal{T}$, the UAVs take actions simultaneously. In particular, each UAV j aims at determining its path $\boldsymbol{p}_j$ to its destination along with its optimal transmission power and cell association vector for each location $a \in \mathcal{A}$ along its path $\boldsymbol{p}_j$. Therefore, at each t, UAV j chooses an action $z_j(t) \in \mathcal{Z}_j$ composed of the tuple $\boldsymbol{z}_j(t) = (\boldsymbol{a}_j(t), \widehat{P}_{j,s,a}(t), \boldsymbol{\beta}_{j,s,a}(t))$, where $\boldsymbol{a}_j(t)$={left, right, forward, backward, no movement} corresponds to a fixed step size, $\widetilde{a}_j$, in a given direction. $\widehat{P}_{j,s,a}(t) = [\widehat{P}_1, \widehat{P}_2, \cdots, \widehat{P}_O]$ corresponds to O different maximum transmit power levels for each UAV j and $\boldsymbol{\beta}_{j,s,a}(t)$ is the UAV-BS association vector.

For each UAV j, let $\mathcal{L}_j$ be the set of its L_j nearest BSs. The observed network state by UAV j at stage t, $\boldsymbol{v}_j(t) \in \mathcal{V}_j$, is:

$$\boldsymbol{v}_j(t){=}\Big[\{\delta_{j,l,a}(t),\theta_{j,l,a}(t)\}_{l=1}^{L_j},\theta_{j,d_j,a}(t),\{x_j(t),y_j(t)\}_{j\in\mathcal{J}}\Big], \quad (16)$$

where $\delta_{j,l,a}(t)$ is the Euclidean distance from UAV j at location a to BS l at stage t, $\theta_{j,l,a}$ is the orientation angle in the xy-plane from UAV j at location a to BS l defined as $\tan^{-1}(\Delta y_{j,l}/\Delta x_{j,l})$ [19] where $\Delta y_{j,l}$ and $\Delta x_{j,l}$ correspond to the difference in the x and y coordinates of UAV j and BS l, $\theta_{j,d_j,a}$ is the orientation angle in the xy-plane from UAV j at location a to its destination d_j defined as $\tan^{-1}(\Delta y_{j,d_j}/\Delta x_{j,d_j})$, and $\{x_j(t),y_j(t)\}_{j\in\mathcal{J}}$ are the horizonal coordinates of all UAVs at stage t. For our model, we consider different range intervals for mapping each of the orientation angle and distance values, respectively, into different states.

Moreover, based on the optimization problem defined in (6)-(15) and by incorporating the Lagrangian penalty method into the utility function definition for the SINR constraint (13), the resulting utility function for UAV j at stage t, $u_j(\boldsymbol{v}_j(t), \boldsymbol{z}_j(t), \boldsymbol{z}_{-j}(t))$, will be given by:

$$u_j(\boldsymbol{v}_j(t), \boldsymbol{z}_j(t), \boldsymbol{z}_{-j}(t)) \\ = \begin{cases} \Phi(\boldsymbol{v}_j(t),\boldsymbol{z}_j(t),\boldsymbol{z}_{-j}(t))+C, \text{ if } \delta_{j,d_j,a}(t) < \delta_{j,d_j,a'}(t-1), \\ \Phi(\boldsymbol{v}_j(t),\boldsymbol{z}_j(t),\boldsymbol{z}_{-j}(t)), \text{ if } \delta_{j,d_j,a}(t) = \delta_{j,d_j,a'}(t-1), \\ \Phi(\boldsymbol{v}_j(t),\boldsymbol{z}_j(t),\boldsymbol{z}_{-j}(t))\text{-}C, \text{ if } \delta_{j,d_j,a}(t) > \delta_{j,d_j,a'}(t-1), \end{cases} \tag{17}$$

where $\Phi(\boldsymbol{v}_j(t),\boldsymbol{z}_j(t),\boldsymbol{z}_{-j}(t))$ is defined as:

$$\Phi(\boldsymbol{v}_j(t),\boldsymbol{z}_j(t),\boldsymbol{z}_{-j}(t)) = -\varrho' \sum_{c=1}^{C_{j,s}(t)} \sum_{r=1,r\neq s}^{S} \frac{\widehat{P}_{j,s,a}(\boldsymbol{v}_j(t)) h_{j,r,c,a}(t)}{C_{j,s}(t)} \\ - \phi' \tau_{j,s,a}(\boldsymbol{v}_j(t),\boldsymbol{z}_j(t),\boldsymbol{z}_{-j}(t)) \\ - \varsigma(\min(0, \sum_{c=1}^{C_{j,s}(t)} \Gamma_{j,s,c,a}(\boldsymbol{v}_j(t),\boldsymbol{z}_j(t),\boldsymbol{z}_{-j}(t)) - \overline{\Gamma}_j))^2, \tag{18}$$

subject to (7)-(12), (14) and (15). ς is the penalty coefficient for (13) and C is a constant parameter. a' and a are the locations of UAV j at $(t-1)$ and t where $\delta_{j,d_j,a}$ is the distance between UAV j and its destination d_j.

B. Equilibrium Analysis

For our game $\mathcal{G}$, we are interested in studying the subgame perfect Nash equilibrium (SPNE) in behavioral strategies. An SPNE is a profile of strategies which induces a Nash equilibrium (NE) on every subgame of the original game. Moreover, a *behavioral strategy* allows each UAV to assign independent probabilities to the set of actions at each network state that is independent across different network states. Here, note that there always exists at least one SPNE for any finite horizon extensive game with perfect information [Kuhn's Theorem] [18]. Let $\boldsymbol{\pi}_j(\boldsymbol{v}_j(t)) = (\pi_{j,z_1}(\boldsymbol{v}_j(t)), \pi_{j,z_2}(\boldsymbol{v}_j(t)), \cdots, \pi_{j,\boldsymbol{z}_{|Z_j|}}(\boldsymbol{v}_j(t))) \in \Pi_j$ be the behavioral strategy of UAV j at state $\boldsymbol{v}_j(t)$ and let $\Delta(\mathcal{Z})$ be the set of all probability distributions over the action space $\mathcal{Z}$. Next, we define the notion of an SPNE.

Definition 1. A behavioral strategy $(\boldsymbol{\pi}_1^*(\boldsymbol{v}_j(t)), \cdots, \boldsymbol{\pi}_J^*(\boldsymbol{v}_j(t))) = (\boldsymbol{\pi}_j^*(\boldsymbol{v}_j(t)), \boldsymbol{\pi}_{-j}^*(\boldsymbol{v}_j(t)))$ constitutes a *subgame perfect Nash equilibrium* if, $\forall j \in \mathcal{J}$, $\forall t \in \mathcal{T}$ and $\forall \boldsymbol{\pi}_j(\boldsymbol{v}_j(t)) \in \Delta(\mathcal{Z})$, $\overline{u}_j(\boldsymbol{\pi}_j^*(\boldsymbol{v}_j(t)), \boldsymbol{\pi}_{-j}^*(\boldsymbol{v}_j(t))) \geq \overline{u}_j(\boldsymbol{\pi}_j(\boldsymbol{v}_j(t)), \boldsymbol{\pi}_{-j}^*(\boldsymbol{v}_j(t)))$.

Therefore, the objective of each UAV j for each state $\boldsymbol{v}_j(t)$ and stage t is to maximize its expected sum of discounted rewards, which is computed as the summation of the immediate reward for a given state along with the expected discounted utility of the next states, as given below:

$$\overline{u}(\boldsymbol{v}_j(t), \boldsymbol{\pi}_j(\boldsymbol{v}_j(t)), \boldsymbol{\pi}_{\text{-}j}(\boldsymbol{v}_j(t))) \\ = \mathbb{E}_{\boldsymbol{\pi}_j(t)} \left\{ \sum_{l=0}^{\infty} \gamma^l u_j(\boldsymbol{v}_j(t+l), \boldsymbol{z}_j(t+l), \boldsymbol{z}_{\text{-}j}(t+l)) | \boldsymbol{v}_{j,0} = \boldsymbol{v}_j \right\} \\ = \sum_{\boldsymbol{z}\in\mathcal{Z}} \sum_{l=0}^{\infty} \gamma^l u_j(\boldsymbol{v}_j(t+l), \boldsymbol{z}_j(t+l), \boldsymbol{z}_{\text{-}j}(t+l)) \prod_{j=1}^{J} \pi_{j,z_j}(\boldsymbol{v}_j(t+l)), \tag{19}$$

where $\gamma^l \in (0,1)$ is a discount factor for delayed rewards and $\mathbb{E}_{\boldsymbol{\pi}_j(\boldsymbol{v}_j(t))}$ denotes an expectation over trajectories of states and actions, in which actions are selected according to $\boldsymbol{\pi}_j(\boldsymbol{v}_j(t))$. Here, $\boldsymbol{u}_j$ is the short-term reward for being in state $\boldsymbol{v}_j$ and $\overline{\boldsymbol{u}}_j$ is the expected long-term total reward from state $\boldsymbol{v}_j$ onwards.

To find the SPNE, each UAV must have full knowledge of the future reward functions at each information set and thus for all future network states. This in turn necessitates knowledge of all possible future actions of all UAVs in the network and becomes challenging as the number of UAVs increases. To address this challenge, we rely on deep recurrent neural networks (RNNs) [20]. In essence, RNNs exhibit dynamic temporal behavior and are characterized by their adaptive memory that enables them to store necessary previous state information to predict future actions. On the other hand, deep neural networks are capable of dealing with large datasets. Therefore, next, we develop a novel deep RL based on ESNs, a special kind of RNN, for solving the SPNE of our game $\mathcal{G}$.

IV. Deep Reinforcement Learning for Online Path Planning and Resource Management

In this section, we first introduce a deep ESN-based architecture that allows UAVs to store previous states whenever needed while being able to learn future network states. Then, we propose an RL algorithm based on the proposed deep ESN architecture to learn an SPNE for our proposed game.

A. Deep ESN Architecture

ESNs are a new type of RNNs with feedback connections that belong to the family of reservoir computing (RC) [20]. An ESN is composed of an input weight matrix $\boldsymbol{W}_{\text{in}}$, a recurrent matrix $\boldsymbol{W}$, and an output weight matrix $\boldsymbol{W}_{\text{out}}$. Because only the output weights are altered, ESN training is typically quick and computationally efficient compared to training other RNNs. Moreover, multiple non-linear reservoir layers can be stacked on top of each other resulting in a *deep ESN architecture*. Deep ESNs exploit the advantages of a hierarchical temporal feature representation at different levels of abstraction while preserving the RC training efficiency. They have the ability to learn data representations at different levels of abstraction, hence disentangling the difficulties in modeling complex tasks by representing them in terms of simpler ones hierarchically. Let $N_{j,R}^{(n)}$ be the number of internal units of the reservoir of UAV j at layer n, $N_{j,U}$ be the external input dimension of UAV j and $N_{j,L}$ be the number of layers in the stack for UAV j. Next, we define the following ESN components:

- $\boldsymbol{v}_j(t) \in \mathbb{R}^{N_{j,U}}$ the external input of UAV j at stage t which effectively corresponds to the current network state,
- $\boldsymbol{x}_j^{(n)}(t) \in \mathbb{R}^{N_{j,R}^{(n)}}$ as the state of the reservoir of UAV j at layer n at stage t,
- $\boldsymbol{W}_{j,\text{in}}^{(n)}$ as the input-to-reservoir matrix of UAV j at layer n, where $\boldsymbol{W}_{j,\text{in}}^{(n)} \in \mathbb{R}^{N_{j,R}^{(n)} \times N_{j,U}}$ for $n = 1$, and $\boldsymbol{W}_{j,\text{in}}^{(n)} \in \mathbb{R}^{N_{j,R}^{(n)} \times N_{j,R}^{(n-1)}}$ for $n > 1$,
- $\boldsymbol{W}_j^{(n)} \in \mathbb{R}^{N_{j,R}^{(n)} \times N_{j,R}^{(n)}}$ as the recurrent reservoir weight matrix for UAV j at layer n,
- $\boldsymbol{W}_{j,\text{out}} \in \mathbb{R}^{|\mathcal{Z}_j| \times (N_{j,U} + \sum_n N_{j,R}^{(n)})}$ as the reservoir-to-output matrix of UAV j for the n^{th} layer only.

The objective of the deep ESN architecture is to approximate a function $\boldsymbol{F}_j = (F_j^1, F_j^2, \cdots, F_j^{N_{j,L}})$ for learning an SPNE for each UAV j at each stage t. For each $n = 1, 2, \cdots, N_{j,L}$, the function $F_j^{(n)}$ describes the evolution of the state of the reservoir at layer n, i.e., $\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{(n)}}(t) = F_j^{(n)}(\boldsymbol{v}_j(t), \boldsymbol{x}_j^{(n)}(t-1))$ for $n = 1$ and $\boldsymbol{x}_{\boldsymbol{j}}^{\boldsymbol{(n)}}(t) = F_j^{(n)}(\boldsymbol{x}_j^{(n-1)}(t), \boldsymbol{x}_j^{(n)}(t-1))$ for $n >$

1. $\boldsymbol{W}_{j,\text{out}}$ and $\boldsymbol{x}_j^{(n)}(t)$ are initialized to zero while $\boldsymbol{W}_{j,\text{in}}^{(n)}$ and $\boldsymbol{W}_j^{(n)}$ are randomly generated. Note that although the dynamic reservoir is initially generated randomly, it is combined later with the external input, $\boldsymbol{v}_j(t)$, in order to store the network states and with the trained output matrix, $\boldsymbol{W}_{j,\text{out}}$, so that it can approximate the reward function. Moreover, the spectral radius of $\boldsymbol{W}_j^{(n)}$ (i.e., the largest eigenvalue in absolute value), $\rho_j^{(n)}$, must be strictly smaller than 1 to guarantee the stability of the reservoir [21]. In fact, the value of $\rho_j^{(n)}$ is related to the variable memory length of the reservoir that enables the proposed deep ESN framework to store necessary previous state information, with larger values of $\rho_j^{(n)}$ resulting in longer memory length.

We next define the deep ESN components: the input and reward functions. For each deep ESN of UAV j, we distinguish between two types of inputs: external input, $\boldsymbol{v}_j(t)$, that is fed to the first layer of the deep ESN and corresponds to the current state of the network and input that is fed to all other layers for $n > 1$. For our proposed deep ESN, the input to any layer $n > 1$ at stage t corresponds to the state of the previous layer, $\boldsymbol{x}_j^{(n-1)}(t)$. Define $\widetilde{u}_j(\boldsymbol{v}_j(t), \boldsymbol{z}_j(t), \boldsymbol{z}_{-j}(t)) = u_j(\boldsymbol{v}_j(t), \boldsymbol{z}_j(t), \boldsymbol{z}_{-j}(t)) \prod_{j=1}^{J} \pi_{j,z_j}(\boldsymbol{v}_j(t))$ as the expected value of the instantaneous utility function $u_j(\boldsymbol{v}_j(t), \boldsymbol{z}_j(t), \boldsymbol{z}_{-j}(t))$ in (17) for UAV j at stage t. Therefore, the reward that UAV j obtains from action $\boldsymbol{z}_j$ at a given network state $\boldsymbol{v}_j(t)$:

$$\begin{aligned} & r_j(\boldsymbol{v}_j(t), \boldsymbol{z}_j(t), \boldsymbol{z}_{-j}(t)) \\ &= \begin{cases} \widetilde{u}_j(\boldsymbol{v}_j(t),\boldsymbol{z}_j(t),\boldsymbol{z}_{\text{-}j}(t)), \text{if UAV } j \text{ reaches } d_j, \\ \widetilde{u}_j(\boldsymbol{v}_j(t),\boldsymbol{z}_j(t),\boldsymbol{z}_{\text{-}j}(t)) \\ \quad +\gamma \max_{\boldsymbol{z}_j \in \mathcal{Z}_j} \boldsymbol{W}_{j,\text{out}}(\boldsymbol{z}_j(t{+}1),t{+}1)[\boldsymbol{v}'_j(t), \boldsymbol{x}_j'^{(1)}(t), \\ \quad \boldsymbol{x}_j'^{(2)}(t), \cdots, \boldsymbol{x}_j'^{(n)}(t)], \text{otherwise.} \end{cases} \end{aligned} \tag{20}$$

Here, $\boldsymbol{v}'_j(t{+}1)$ and $\boldsymbol{x}_j'^{(n)}(t)$, correspond, respectively, to the next network state and reservoir state of layer (n), at stage $(t+1)$, upon taking actions $\boldsymbol{z}_j(t)$ and $\boldsymbol{z}_{-j}(t)$ at stage t.

B. Update Rule Based on Deep ESN

We now introduce the deep ESN's update phase that each UAV uses to store and estimate the reward function of each path and resource allocation scheme at a given stage t. In particular, we consider leaky integrator reservoir units [22] for updating the state transition functions $\boldsymbol{x}_j^{(n)}(t)$ at stage t. Therefore, the state transition function of the first layer $\boldsymbol{x}_j^{(1)}(t)$ will be:

$$\begin{aligned} \boldsymbol{x}_j^{(1)}(t) &= (1-\omega_j^{(1)})\boldsymbol{x}_j^{(1)}(t-1) \\ &\quad + \omega_j^{(1)} \tanh(\boldsymbol{W}_{j,\text{in}}^{(1)} \boldsymbol{v}_j(t) + \boldsymbol{W}_j^{(1)} \boldsymbol{x}_j^{(1)}(t-1)), \end{aligned} \tag{21}$$

where $\omega_j^{(n)} \in [0,1]$ is the leaking parameter at layer n for UAV j which relates to the speed of the reservoir dynamics in response to the input, with larger values of $\omega_j^{(n)}$ resulting in a faster response of the corresponding n-th reservoir to the input. The state transition of UAV j, $\boldsymbol{x}_j^{(n)}(t)$, for $n > 1$ is given by:

$$\begin{aligned} \boldsymbol{x}_j^{(n)}(t) &= (1-\omega_j^{(n)})\boldsymbol{x}_j^{(n)}(t-1) \\ &\quad + \omega_j^{(n)} \tanh(\boldsymbol{W}_{j,\text{in}}^{(n)} \boldsymbol{x}_j^{(n-1)}(t) + \boldsymbol{W}_j^{(n)} \boldsymbol{x}_j^{(n)}(t-1)), \end{aligned} \tag{22}$$

The output $y_j(t)$ of the deep ESN at stage t is used to estimate the reward of each UAV j based on the current adopted action $\boldsymbol{z}_j(t)$ and $\boldsymbol{z}_{-j}(t)$ of UAV j and other UAVs $(-j)$, respectively, for the current network state $\boldsymbol{v}_j(t)$ after training $\boldsymbol{W}_{j,\text{out}}$. It can be computed as:

$$\begin{aligned} y_j(\boldsymbol{v}_j(t), \boldsymbol{z}_j(t)) = \boldsymbol{W}_{j,\text{out}}(\boldsymbol{z}_j(t), t)[\boldsymbol{v}_j(t), \boldsymbol{x}_j^{(1)}(t), \\ \boldsymbol{x}_j^{(2)}(t), \cdots, \boldsymbol{x}_j^{(n)}(t)]. \end{aligned} \tag{23}$$

We adopt a temporal difference RL approach for training the output matrix $W_{j,\text{out}}$ of the deep ESN architecture. In particular, we employ a linear gradient descent approach using the reward error signal, given by the following update rule [23]:

$$\begin{aligned} \boldsymbol{W}_{j,\text{out}}(\boldsymbol{z}_j(t),t{+}1) {=} \boldsymbol{W}_{j,\text{out}}(\boldsymbol{z}_j(t),t) {+} \lambda_j(r_j(\boldsymbol{v}_j(t),\boldsymbol{z}_j(t),\boldsymbol{z}_{\text{-}j}(t)) \\ - y_j(\boldsymbol{v}_j(t),\boldsymbol{z}_j(t)))[\boldsymbol{v}_j(t),\boldsymbol{x}_j^{(1)}(t),\boldsymbol{x}_j^{(2)}(t),\cdots,\boldsymbol{x}_j^{(n)}(t)]^T. \end{aligned} \tag{24}$$

Here, note that the objective of each UAV is to minimize the value of the error function $e_j(\boldsymbol{v}_j(t)) = |r_j(\boldsymbol{v}_j(t),\boldsymbol{z}_j(t),\boldsymbol{z}_{\text{-}j}(t)) - y_j(\boldsymbol{v}_j(t),\boldsymbol{z}_j(t))|$.

C. Proposed Deep RL Algorithm

Based on the proposed deep ESN architecture and update rule, we next introduce a multi-agent deep RL framework that the UAVs can use to learn an SPNE in behavioral strategies for the game $\mathcal{G}$. The algorithm is divided into two phases: *training and testing*. In the former, UAVs are trained offline before they become active in the network using the architecture of Subsection IV-A. The testing phase corresponds to the actual execution of the algorithm after which the weights of $\boldsymbol{W}_{j,\text{out}}, \forall j \in \mathcal{J}$ have been optimized and is implemented on each UAV for execution during run time.

During the training phase, each UAV aims at optimizing its output weight matrix $\boldsymbol{W}_{j,\text{out}}$ such that the value of the error function $e_j(\boldsymbol{v}_j(t))$ at each stage t is minimized. In particular, the training phase is composed of multiple iterations, each consisting of multiple rounds, i.e., the number of steps required for all UAVs to reach their corresponding destinations d_j. At each round, UAVs face a tradeoff between playing the action associated with the highest expected utility, and trying out all their actions to improve their estimates of the reward function in (20). This in fact corresponds to the exploration and exploitation tradeoff, in which UAVs need to strike a balance between exploring their environment and exploiting the knowledge accumulated through such exploration [24]. Therefore, we adopt the ϵ-greedy policy in which UAVs choose the action that yields the maximum utility value with a probability of $1-\epsilon+\frac{\epsilon}{|\mathcal{Z}_j|}$ while exploring randomly other actions with a probability of $\frac{\epsilon}{|\mathcal{A}_j|}$. The strategy over the action space can thus be defined as:

$$\pi_{j,z_j}(\boldsymbol{v}_j(t)) = \begin{cases} 1-\epsilon+\frac{\epsilon}{|\mathcal{Z}_j|}, \ \text{argmax}_{z_j \in \mathcal{Z}_j} y_j\left(\boldsymbol{v}_j(t), \boldsymbol{z}_j(t)\right), \\ \frac{\epsilon}{|\mathcal{Z}_j|}, \ \text{otherwise.} \end{cases} \tag{25}$$

Based on the selected action $\boldsymbol{z}_j(t)$, each UAV j updates its location, cell association, and transmission power level and computes its reward function according to (20). To determine the next network state, each UAV j broadcasts its selected action to all other UAVs in the network. Then, each UAV j updates its state transition vector $\boldsymbol{x}_j^{(n)}(t)$ for each layer (n) of the deep ESN architecture according to (21) and (22). The output y_j at stage t is then updated based on (23). Finally, the weights of the output matrix $\boldsymbol{W}_{j,\text{out}}$ of each UAV j are updated based on the linear gradient descent update rule given in (24). Note that, a UAV stops taking any actions once it has reached its destination. The convergence complexity of this deep RL algorithm is $O(J)$. A summary of the training phase is given in Algorithm 1.

Algorithm 1: Training phase of the proposed deep RL algorithm

```
Initialization:
$\boldsymbol{\pi}_{j,z_j}(\boldsymbol{v}_j(t)) = \frac{1}{|\mathcal{A}_j|} \forall t \in T, z_j \in \mathcal{Z}_j$, $y_j(\boldsymbol{v}_j(t), \boldsymbol{z}_j(t)) = 0$, $\boldsymbol{W}_{j,\text{in}}^{(n)}$, $\boldsymbol{W}_j^{(n)}$, $\boldsymbol{W}_{j,\text{out}}$.

for The number of training iterations do
  while At least one UAV $j$ has not reached its destination $d_j$, do
    for all UAVs $j$ (in a parallel fashion) do
      Input: Each UAV $j$ receives an input $\boldsymbol{v}_j(t)$ based on (16).
      Step 1: Action selection
      Each UAV $j$ selects a random action $\boldsymbol{z}_j(t)$ with probability $\epsilon$,
      Otherwise, UAV $j$ selects $\boldsymbol{z}_j(t) = \text{argmax}_{\boldsymbol{z}_j \in \mathcal{Z}_j} y_j(\boldsymbol{v}_j(t), \boldsymbol{z}_j(t))$.
      Step 2: Location, cell association and transmit power update
      Each UAV $j$ updates its location, cell association and transmission power level based on the selected action $\boldsymbol{z}_j(t)$.
      Step 3: Reward computation
      Each UAV $j$ computes its reward values based on (20).
      Step 4: Action broadcast
      Each UAV $j$ broadcasts its selected action $\boldsymbol{z}_j(t)$ to all other UAVs.
      Step 5: Deep ESN update
      - Each UAV $j$ updates the state transition vector $\boldsymbol{x}_j^{(n)}(t)$ for each layer $(n)$ of the deep ESN architecture based on (21) and (22).
      - Each UAV $j$ computes its output $y_j(\boldsymbol{v}_j(t), \boldsymbol{z}_j(t))$ based on (23).
      - The weights of the output matrix $\boldsymbol{W}_{j,\text{out}}$ of each UAV $j$ are updated based on the linear gradient descent update rule given in (24).
    end for
  end while
end for
```

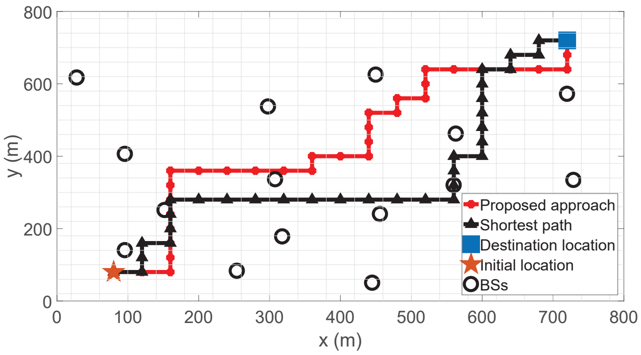

Fig. 1. Path of a UAV for our approach and shortest path scheme.

TABLE I
Performance assessment for one UAV

	# of steps	delay (ms)	average rate per UE (Mbps)
Proposed approach	32	6.5	0.95
Shortest path	32	12.2	0.76

Meanwhile, the testing phase corresponds to the actual execution of the algorithm. In this phase, each UAV chooses its action greedily for each state $\boldsymbol{v}_j(t)$, i.e., $\text{argmax}_{\boldsymbol{z}_j \in \mathcal{Z}_j} y_j(\boldsymbol{v}_j(t), \boldsymbol{z}_j(t))$, and updates its location, cell association, and transmission power level accordingly. Each UAV then broadcasts its selected action and updates its state transition vector $\boldsymbol{x}_j^{(n)}(t)$ for each layer n of the deep ESN architecture based on (21) and (22). It is important to note that, upon convergence, the convergence strategy profile corresponds to an SPNE of game $\mathcal{G}$ due to the fact that for any finite game of perfect information, any backward induction solution is an SPNE [18].

V. Simulation Results and Analysis

For our simulations, we consider an 800 m × 800 m square area divided into 40 m × 40 m grid areas, in which we randomly uniformly deploy 15 BSs. All statistical results are averaged over several independent testing iterations during which the initial locations and destinations of the UAVs and the location of the BSs and the ground UEs are randomized. The maximum transmit power for each UAV is discretized into 5 equally separated levels. The multipath fading is considered to be uncorrelated Rician fading with parameter $\widehat{K} = 1.59$. The external input of the deep ESN architecture, $\boldsymbol{v}_j(t)$, is a function of the number of UAVs and thus the number of hidden nodes per layer, $N_{j,R}^{(n)}$, varies with the number of UAVs. For instance, $N_{j,R}^{(n)} = 12$ and 6 for $n = 1$ and 2, respectively, for a network size of 1 and 2 UAVs, and 20 and 10 for a network size of 3, 4, and 5 UAVs. Table II summarizes the main simulation parameters.

TABLE II
SYSTEM PARAMETERS

Parameters	Values	Parameters	Values
UAV max transmit power $(\overline{P}_j)$	20 dBm	SINR threshold $(\overline{\Gamma}_j)$	-3 dB
UE transmit power $(\widehat{P}_q)$	20 dBm	Learning rate (λ_j)	0.01
Noise power spectral density (N_0)	-174 dBm/Hz	RB bandwidth (B_c)	180 kHz
Total bandwidth (B)	20 MHz	# of interferers (L)	2
Packet arrival rate $(\lambda_{j,s})$	(0,1)	Packet size (ν)	2000 bits
Carrier frequency (f_c)	2 GHz	Discount factor (γ)	0.7
# of hidden layers	2	Step size $(\widetilde{a}_j)$	40 m
Leaky parameter/layer $(\omega_j^{(n)})$	0.99, 0.99	ϵ	0.3

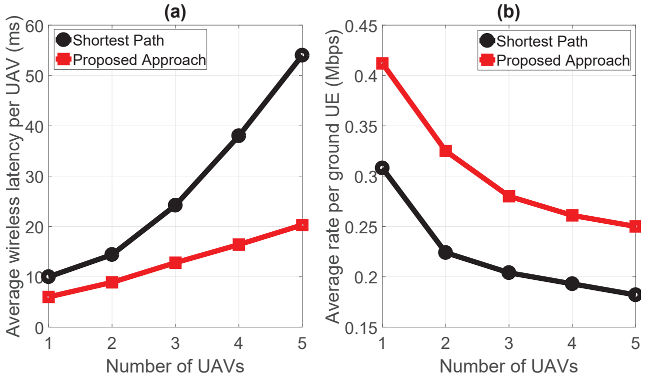

Fig. 2. Performance assessment of the proposed approach in terms of average (a) wireless latency per UAV and (b) rate per ground UE as compared to the shortest path approach, for different number of UAVs.

TABLE III
The required number of steps for all UAVs to reach their corresponding destinations based on our proposed approach and that of the shortest path scheme for different number of UAVs

# of steps	1 UAV	2 UAVs	3 UAVs	4 UAVs	5 UAVs
Proposed approach	4	4	6	7	8
Shortest path	4	4	6	6	7

Fig. 1 shows a snapshot of the path of a single UAV resulting from our approach and from a shortest path scheme. Unlike our proposed scheme which accounts for other wireless metrics during path planning, the objective of the UAVs in the shortest path scheme is to merely reach their destinations with the minimum number of steps. Table I presents the performance results for the paths shown in Fig. 1. From Fig. 1, we can see that, for our proposed approach, the UAV selects a path away from the congested area of BSs while maintaining proximity to its serving BS in a way that would minimize the steps required to reach its destination. This in turn minimizes the interference level it causes on the ground UEs and its wireless latency (Table I). From Table I, we can see that our proposed approach achieves 25% increase in the average rate per ground UE and 47% decrease in the wireless latency as compared to the shortest path, while requiring the same number of steps to reach its destination.

Fig. 2 compares the average values of the (a) wireless latency per UAV and (b) rate per ground UE for our proposed approach and the baseline shortest path scheme. Moreover, Table III compares the required number of steps for all UAVs to reach their corresponding destinations for the scenarios presented in Fig. 2. From Fig. 2 and Table III, we can see that, compared to the shortest path scheme, our approach achieves better wireless latency per UAV and rate per ground UE for different numbers

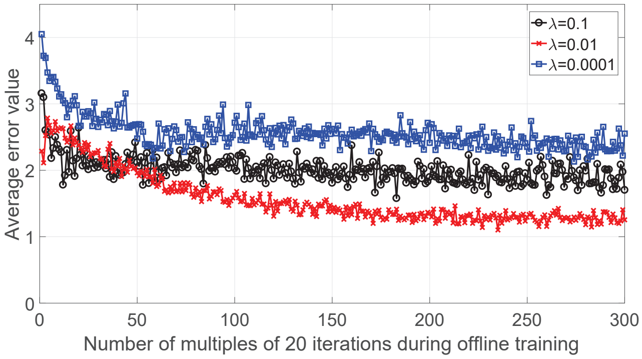

Fig. 3. Effect of the learning rate on the convergence of offline training.

of UAVs while requiring a number of steps that is comparable to the baseline. In fact, our scheme provides a better tradeoff between energy efficiency, wireless latency, and achievable data rate of the ground UEs compared to the shortest path scheme. For instance, for 5 UAVs, our scheme achieves a 37% increase in the average achievable rate per ground UE, 62% decrease in the average wireless latency per UAV, and 14% decrease in energy efficiency. Indeed, one can adjust the multi-objective weights of our utility function based on the rate requirements of the ground network, power limitation of the UAVs, and the maximum tolerable wireless latency of the UAVs. Moreover, Fig. 2 shows that, as the number of UAVs increases, the average delay per UAV increases and the average rate per ground UE decreases, for our scheme as well as that of the shortest path scheme. This in fact results from the LoS link between the UAVs and the BSs which in turn increases the interference level on the ground UEs and other UAVs.

Fig. 3 shows the average of the error function $e_j(\boldsymbol{v}_j(t))$ resulting from the offline training phase as a function of a multiple of 20 iterations while considering different values for the learning rate (λ). The learning rate determines the step size the algorithm takes to reach the optimal solution and, thus, it impacts the convergence rate of our proposed framework. From Fig. 3, we can see that small values of the learning rate, i.e., $\lambda = 0.0001$, result in a slow speed of convergence. On the other hand, for large values of the learning rate, such as $\lambda = 0.1$, the error function decays fast for the first few iterations but then remains constant. Here, note that $\lambda = 0.1$ does not lead to convergence during the testing phase, but $\lambda = 0.0001$ and $\lambda = 0.01$ result in convergence, though requiring a different number of training iterations. In fact, a large learning rate can cause the algorithm to diverge from the optimal solution. This is because too large initial learning rates will decay the loss function faster and thus make the model get stuck at a particular region of the optimization space instead of better exploring it. Clearly, our proposed framework achieves better performance for $\lambda = 0.01$, as compared to smaller and larger values of the learning rate. It is worthwhile noting here that the error function does not reach the value of zero during the training phase. This is due to the fact that, for our approach, we adopt the early stopping technique to avoid overfitting which occurs when the training error decreases at the expense of an increase in the value of the test error [20].

VI. Conclusion

In this paper, we have proposed a novel interference-aware path planning scheme for a multi-UAV network. We have formulated the problem as a noncooperative game in which the UAVs are the players. To solve the game, we have proposed a deep RL algorithm based ESN cells which is guaranteed to reach an SPNE if converged. Simulation results have shown that the proposed approach achieves better wireless latency per UAV and rate per ground UE while requiring a number of steps that is comparable to the shortest path scheme.

References

[1] 3GPP, "3GPP: study on enhanced support for aerial vehicles," March 2017. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3231

[2] Qualcomm, "Paving the path to 5G: Optimizing commercial LTE networks for drone communication," [Online], Sept. 2016. [Online]. Available: https://www.qualcomm.com/news/onq/2016/09/06/paving-path-5g-optimizing-commercial-lte-networks-drone-communication

[3] B. V. der Bergh, A. Chiumento, and S. Pollin, "LTE in the sky: Trading off propagation benefits with interference costs for aerial nodes," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 44–50, May 2016.

[4] X. Lin, V. Yajnanarayana, S. Muruganathan, S. Gao, and H. Asplund, "The sky is not the limit: LTE for unmanned aerial vehicles," *arXiv:1707.07534*, July 2017.

[5] M. Azari, F. Rosas, A. Chiumento, and S. Pollin, "Coexistence of terrestrial and aerial users in cellular networks," in *Proc. of IEEE Global communications conference (GLOBECOM) workshops*. Singapore, Dec. 2017.

[6] T. Andre, K. Hummel, A. Schoellig, E. Yanmaz, M. Asadpour, C. Bettstetter, P. Grippa, H. Hellwagner, S. Sand, and S. Zhang, "Application-driven design of aerial communication networks," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 129–137, May 2014.

[7] U. Challita and W. Saad, "Network formation in the sky: Unmanned aerial vehicles for multi-hop wireless backhauling," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*. Singapore, Dec. 2017.

[8] J. Yoon, Y. Jin, N. Batsoyol, and H. Lee, "Adaptive path planning of UAVs for delivering delay-sensitive information to ad-hoc nodes," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*. San Francisco, CA, USA, Mar. 2017.

[9] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, June 2017.

[10] M. Messous, S. Senouci, and H. Sedjelmaci, "Network connectivity and area coverage for UAV fleet mobility model with energy constraint," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*. Doha, Qatar, Apr. 2016.

[11] X. Wang, A. Chowdhery, and M. Chiang, "Networked drone cameras for sports streaming," in *Proc. of International Conference on Distributed Computing Systems (ICDCS)*. Atlanta, Georgia, USA, June 2017.

[12] Y. Zeng, R. Zhang, and T. Lim, "Throughput maximization for UAV-enabled mobile relaying systems," *IEEE Transactions on Communications*, vol. 64, no. 12, pp. 4983–4996, Dec. 2016.

[13] M. Bekhti, M. Abdennebi, N. Achir, and K. Boussetta, "Path planning of unmanned aerial vehicles with terrestrial wireless network tracking," in *Proc. of Wireless Days*. Toulouse, France, May 2016.

[14] A. Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*. Austin, TX, USA, Dec. 2014.

[15] U. Mengali and A. D'Andrea, *Synchronization Techniques for Digital Receivers*, Plenum Press, Ed., New York, 1997.

[16] 3GPP TR 25.942 v2.1.3, "3rd generation partnership project; technical specification group (TSG) RAN WG4; RF system scenarios," Tech. Rep., 2000.

[17] D. Bertsekas and R. Gallager, *Data Networks*. Prentice Hall, Mar. 1992.

[18] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjorungnes, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*. Cambridge University Press, 2012.

[19] W. Kwon, I. Suh, S. Lee, and Y. Cho, "Fast reinforcement learning using stochastic shortest paths for a mobile robot," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Diego, CA, USA, Nov. 2007.

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

[21] C. Gallicchio and A. Micheli, "Echo state property of deep reservoir computing networks," *Cognitive Computation*, vol. 9, pp. 337–350, May 2017.

[22] H. Jaeger, M. Lukosevicius, D. Popovici, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural Networks*, vol. 20, no. 3, pp. 335–352, 2007.

[23] I. Szita and A. L. V. Gyenes, *Reinforcement Learning with Echo State Networks*. Springer, Berlin, Heidelberg, 2006, vol. 4131.

[24] R. Sutton and A. Barto, *Introduction to Reinforcement Learning*, 1998.