

Interference Management for Cellular-Connected UAVs: A Deep Reinforcement Learning Approach

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

In this paper, an interference-aware path planning scheme for a network of cellular-connected unmanned aerial vehicles (UAVs) is proposed. In particular, each UAV aims at achieving a tradeoff between maximizing energy efficiency and minimizing both 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 introduced 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 proposed algorithm is shown to reach a subgame perfect Nash equilibrium (SPNE) upon convergence. Moreover, an upper and lower bound for the altitude of the UAVs is derived thus reducing the computational complexity of the proposed algorithm. Simulation results show that the proposed scheme achieves better wireless latency per UAV and rate per ground user (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. The results also show that the optimal altitude of the UAVs varies based on the ground network density and the UE data rate requirements and plays a vital role in minimizing the interference level on the ground UEs as well as the wireless transmission delay of the UAV.

Index Terms

Unmanned aerial vehicles (UAV); echo state network (ESN); deep learning; deep reinforcement learning; game theory; path planning

The work of W. Saad was supported by the U.S. National Science Foundation under Grant CNS-1836802 and Grant IIS-1633363.

I. INTRODUCTION

Cellular-connected unmanned aerial vehicles (UAVs) will be an integral component of future wireless networks as evidenced by recent interest from academia, industry, and 3GPP standardizations [1]–[6]. Unlike current wireless UAV connectivity that relies on short-range communication range (e.g., WiFi, bluetooth, and radio waves), cellular-connected UAVs allow beyond line-of-sight control, low latency, real time communication, robust security, and ubiquitous coverage. Such *cellular-connected UAV-user equipments (UEs)* will thus 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 for the UAV-UEs. 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]–[9], the performance analysis of cellular-connected UAV-UEs (*short-handed hereinafter as UAVs*) remains relatively scarce [3]–[5], [10]. For instance, in [3], the authors study the impact of UAVs on the uplink performance of a ground LTE network. Meanwhile, the work in [4] uses measurements and ray tracing simulations to study the airborne connectivity requirements and propagation characteristics of UAVs. The authors in [5] analyze the coverage probability of the downlink of a cellular network that serves both aerial and ground users. In [10], the authors consider a network consisting of both ground and aerial UEs and derive closed-form expressions for the coverage probability of the ground and drone UEs. Nevertheless, this prior art is limited to studying the impact that cellular-connected UAVs have on the ground network. Indeed, the existing literature [3]–[5], [10] does not provide any concrete 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. An online path planning that accounts for wireless metrics is vital and would, in essence, assist in addressing the aforementioned interference challenges along with new improvements in the design of the network, such as 3D frequency resue. Such a path planning scheme allows the UAVs to adapt their movement based on the rate requirements of both aerial UAV-UEs and ground UEs, thus improving the overall network performance. The problem of UAV path planning has

been studied mainly for non-UAV-UE applications [8], [9], [11] with [12] being the only work considering a cellular-connected UAV-UE scenario. 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] 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, the body of work in [8], [9] and [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. On the other hand, the authors in [12] propose a path planning scheme for minimizing the time required by a cellular-connected UAV to reach its destination. Nevertheless, this work is limited to one UAV and does not account for the interference that cellular-connected UAVs cause on the ground network during their mission. Moreover, the work in [12] relies on offline optimization techniques that cannot adapt to the uncertainty and dynamics of a cellular network.

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 cellular-connected UAVs to minimize the interference they cause on the ground network as well as their wireless transmission latency. To realize this, we propose a dynamic 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. For our proposed game, the UAV's cell association vector, trajectory optimization, and transmit power level are closely coupled with each other and their optimal values vary based on the dynamics of the network. Therefore, 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. Consequently, 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. Unlike previous studies which are either centralized or rely on the coordination among UAVs, our approach is based on a self-organizing path planning and resource allocation scheme. 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. We have shown that the proposed algorithm reaches a subgame perfect Nash equilibrium (SPNE) upon convergence. Moreover, upper and lower bounds on the UAVs' altitudes, that guarantee a maximum interference level on the ground network and a maximum wireless transmission delay for the UAV, have been derived. 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. Results also show that each UAV's altitude is a function of the ground network density and the UAV's objective function and is an important factor in achieving the UAV's target.

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 VI.

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, h_j) and $(x_q, y_q, 0)$. All UAVs are assumed to fly at a fixed altitude h_j above the ground (as done in [12], [13]) while the horizontal 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 \mathbf{p}_j for each UAV j . The path \mathbf{p}_j of each UAV j is defined as a sequence of area units $\mathbf{p}_j = (a_1, a_2, \dots, a_l)$ such that $a_1 = o_j$ and $a_l = d_j$. The area size of the discretized area units $(a_1, a_2, \dots, a_A) \in \mathcal{A}$ 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. We assume a constant speed $0 < V_j \leq \hat{V}_j$ for each UAV where \hat{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]:

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

where \hat{f} 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. 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].

For the terrestrial UE-BS links, we consider a Rayleigh fading channel. For a carrier frequency, \hat{f} , 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 $P_{j,s,c,a} = \hat{P}_{j,s,a}/C_{j,s}$ is the transmit power of UAV j at location a to BS s over RB c and $\hat{P}_{j,s,a}$ 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:

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

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 $\hat{\mathbf{P}}$ be the transmission power vector with each element $\hat{P}_{j,s,a} \in [0, \bar{P}_j]$ being the transmission power level of UAV j to its serving BS s at location a where \bar{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 \mathbf{p}_j :

$$\min_{\hat{\mathbf{P}}, \boldsymbol{\alpha}, \boldsymbol{\beta}} \vartheta \sum_{j=1}^J \sum_{s=1}^S \sum_{c=1}^{C_{j,s}} \sum_{a=1}^A \sum_{r=1, r \neq s}^S \frac{\hat{P}_{j,s,a} h_{j,r,c,a}}{C_{j,s}} + \varpi \sum_{j=1}^J \sum_{a=1}^A \sum_{b=1, b \neq a}^A \alpha_{j,a,b} + \phi \sum_{j=1}^J \sum_{s=1}^S \sum_{a=1}^A \beta_{j,s,a} \tau_{j,s,a}, \quad (6)$$

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

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

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

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

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

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

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

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

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

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. ϑ , 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 \mathbf{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 $\hat{P}_{j,s,a} > 0$ only if UAV j visits area a , i.e., $a \in \mathbf{p}_j$ 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 \mathbf{p}_j . (13) guarantees an upper limit, $\bar{\Gamma}_j$, 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 , $a \in \mathbf{p}_j$. This, in turn, ensures successful decoding of the transmitted packets at the serving BS. The value of $\bar{\Gamma}_j$ is application and mission specific. Note that the SINR check at each location a is valid for our problem since we consider small-sized area units. (14) and (15) are 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 \mathbf{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 \mathbf{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 \mathbf{p}_j . Therefore, at each t , UAV j chooses an action $z_j(t) \in \mathcal{Z}_j$ composed of the tuple $z_j(t) = (\mathbf{a}_j(t), \widehat{P}_{j,s,a}(t), \boldsymbol{\beta}_{j,s,a}(t))$, where $\mathbf{a}_j(t) = \{\text{left, right, forward, backward, no movement}\}$ corresponds to a fixed step size, \tilde{a}_j , in a given direction. $\widehat{P}_{j,s,a}(t) = [\widehat{P}_1, \widehat{P}_2, \dots, \widehat{P}_O]$ 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 , $\mathbf{v}_j(t) \in \mathcal{V}_j$, is:

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

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 horizontal 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 , $u_j(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))$, will be given by:

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

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

$$\Phi(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)) = -\vartheta' \sum_{c=1}^{C_{j,s}(t)} \sum_{r=1, r \neq s}^S \frac{\widehat{P}_{j,s,a}(\mathbf{v}_j(t)) h_{j,r,c,a}(t)}{C_{j,s}(t)} - \phi' \tau_{j,s,a}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))$$

$$-\varsigma(\min(0, \sum_{c=1}^{C_{j,s}(t)} \Gamma_{j,s,c,a}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)) - \bar{\Gamma}_j))^2, \quad (18)$$

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 . It is worth noting here that the action space of each UAV j and, thus, the complexity of the proposed game \mathcal{G} increases exponentially when updating the 3D coordinates of the UAVs. Nevertheless, each UAV's altitude must be bounded in order to guarantee an SINR threshold for the UAV and a minimum achievable data rate for the ground UEs. Next, we derive an upper and lower bound for the optimal altitude of any given UAV j based on the proposed utility function in (17). In essence, such bounds are valid for all values of the multi-objective weights ϑ' , ϕ' , and ς .

Theorem 1. For all values of ϑ' , ϕ' , and ς , a given network state $\mathbf{v}_j(t)$, and a particular action $\mathbf{z}_j(t)$, the upper and lower bounds for the altitude of UAV j are, respectively, given by:

$$h_j^{\max}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)) = \max(\chi, \hat{h}_j^{\max}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))), \quad (19)$$

$$h_j^{\min}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)) = \max(\chi, \hat{h}_j^{\min}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))), \quad (20)$$

where χ corresponds to the minimum altitude at which a UAV can fly. $\hat{h}_j^{\max}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))$ and $\hat{h}_j^{\min}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))$ are expressed as:

$$\begin{aligned} \hat{h}_j^{\max}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)) = & \sqrt{\frac{\hat{P}_{j,s,a}(\mathbf{v}_j(t))}{C_{j,s}(t) \cdot \bar{\Gamma}_j \cdot \left(\frac{4\pi\hat{f}}{\hat{c}}\right)^2} \cdot \sum_{c=1}^{C_{j,s}(t)} \frac{g_{j,s,c,a}(t)}{I_{j,s,c}(t) + B_c N_0} - (x_j - x_s)^2 - (y_j - y_s)^2}, \quad (21) \end{aligned}$$

and

$$\hat{h}_j^{\min}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)) = \max_r \hat{h}_{j,r}^{\min}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)), \quad (22)$$

where $\hat{h}_{j,r}^{\min}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))$ is the minimum altitude that UAV j should operate at with respect to a particular neighboring BS r and is expressed as:

$$\begin{aligned} \hat{h}_{j,r}^{\min}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)) = & \sqrt{\frac{\hat{P}_{j,s,a}(\mathbf{v}_j(t)) \cdot \sum_{c=1}^{C_{j,s}(t)} g_{j,r,c,a}(t)}{C_{j,s}(t) \cdot \left(\frac{4\pi\hat{f}}{\hat{c}}\right)^2 \cdot \sum_{c=1}^{C_{j,s}(t)} \bar{I}_{j,r,c,a}} - (x_j - x_r)^2 - (y_j - y_r)^2}, \quad (23) \end{aligned}$$

Proof. See Appendix A. ■

From the above theorem, we can deduce that the optimal altitude of the UAVs is a function of their objective function, location of the ground BSs, network design parameters, and the

interference level from other UEs and UAVs in the network. Therefore, at each time step t , UAV j would adjust its altitude level based on the values of $h_j^{\max}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))$ and $h_j^{\min}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))$ thus adapting to the dynamics of the network. In essence, the derived upper and lower bounds for the optimal altitude of the UAVs allows a reduction of the action space of game \mathcal{G} thus simplifying the process needed for the UAVs to find a solution, i.e., equilibrium, of the game. Next, we analyze the equilibrium point of the proposed game \mathcal{G} .

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 [Selten's Theorem] [20]. Let $\boldsymbol{\pi}_j(\mathbf{v}_j(t)) = (\pi_{j,z_1}(\mathbf{v}_j(t)), \pi_{j,z_2}(\mathbf{v}_j(t)), \dots, \pi_{j,z_{|\mathcal{Z}_j|}}(\mathbf{v}_j(t))) \in \Pi_j$ be the behavioral strategy of UAV j at state $\mathbf{v}_j(t)$ 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 $(\boldsymbol{\pi}_1^*(\mathbf{v}_j(t)), \dots, \boldsymbol{\pi}_J^*(\mathbf{v}_j(t))) = (\boldsymbol{\pi}_j^*(\mathbf{v}_j(t)), \boldsymbol{\pi}_{-j}^*(\mathbf{v}_j(t)))$ constitutes a *subgame perfect Nash equilibrium* if, $\forall j \in \mathcal{J}$, $\forall t \in \mathcal{T}$ and $\forall \boldsymbol{\pi}_j(\mathbf{v}_j(t)) \in \Delta(\mathcal{Z})$, $\bar{u}_j(\boldsymbol{\pi}_j^*(\mathbf{v}_j(t)), \boldsymbol{\pi}_{-j}^*(\mathbf{v}_j(t))) \geq \bar{u}_j(\boldsymbol{\pi}_j(\mathbf{v}_j(t)), \boldsymbol{\pi}_{-j}^*(\mathbf{v}_j(t)))$.

Therefore, each state $\mathbf{v}_j(t)$ and stage t , the goal of each UAV j 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:

$$\begin{aligned} \bar{u}(\mathbf{v}_j(t), \boldsymbol{\pi}_j(\mathbf{v}_j(t)), \boldsymbol{\pi}_{-j}(\mathbf{v}_j(t))) &= \mathbb{E}_{\boldsymbol{\pi}_j(\mathbf{v}_j(t))} \left\{ \sum_{l=0}^{\infty} \gamma^l u_j(\mathbf{v}_j(t+l), \mathbf{z}_j(t+l), \mathbf{z}_{-j}(t+l)) \mid \mathbf{v}_{j,0} = \mathbf{v}_j \right\} \\ &= \sum_{\mathbf{z} \in \mathcal{Z}} \sum_{l=0}^{\infty} \gamma^l u_j(\mathbf{v}_j(t+l), \mathbf{z}_j(t+l), \mathbf{z}_{-j}(t+l)) \prod_{j=1}^J \pi_{j,z_j}(\mathbf{v}_j(t+l)), \quad (24) \end{aligned}$$

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

Here, note that the UAV's cell association vector, trajectory optimization, and transmit power level are closely coupled with each other and their corresponding optimal values vary based on the UAVs' objectives. In a multi-UAV network, each UAV must have full knowledge of the future reward functions at each information set and thus for all future network states in order to find the SPNE. 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) [21]. 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 the 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) [21]. An ESN is composed of an input weight matrix \mathbf{W}_{in} , a recurrent matrix \mathbf{W} , and an output weight matrix \mathbf{W}_{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 can 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:

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

The objective of the deep ESN architecture is to approximate a function $\mathbf{F}_j = (F_j^1, F_j^2, \dots, F_j^{N_{j,L}})$ for learning an SPNE for each UAV j at each stage t . For each $n = 1, 2, \dots, N_{j,L}$, the function $F_j^{(n)}$ describes the evolution of the state of the reservoir at layer n , i.e., $\mathbf{x}_j^{(n)}(t) = F_j^{(n)}(\mathbf{v}_j(t), \mathbf{x}_j^{(n)}(t-1))$ for $n = 1$ and $\mathbf{x}_j^{(n)}(t) = F_j^{(n)}(\mathbf{x}_j^{(n-1)}(t), \mathbf{x}_j^{(n)}(t-1))$ for $n > 1$. $\mathbf{W}_{j,\text{out}}$ and $\mathbf{x}_j^{(n)}(t)$ are initialized to zero while $\mathbf{W}_{j,\text{in}}^{(n)}$ and $\mathbf{W}_j^{(n)}$ are randomly generated. Note that although the dynamic reservoir is initially generated randomly, it is combined later with the external input, $\mathbf{v}_j(t)$, in order to store the network states and with the trained output matrix, $\mathbf{W}_{j,\text{out}}$, so that it can approximate the reward function. Moreover, the spectral radius of $\mathbf{W}_j^{(n)}$ (i.e., the largest eigenvalue in absolute value), $\rho_j^{(n)}$, must be strictly smaller than 1 to guarantee the stability of the reservoir [22]. 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, $\mathbf{v}_j(t)$, 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, $\mathbf{x}_j^{(n-1)}(t)$. Define $\tilde{u}_j(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)) = u_j(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)) \prod_{j=1}^J \pi_{j,z_j}(\mathbf{v}_j(t))$ as the expected value of the instantaneous utility function $u_j(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))$ in (17) for UAV j at stage t . Therefore, the reward that UAV j obtains from action \mathbf{z}_j at a given network state $\mathbf{v}_j(t)$:

$$r_j(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)) = \begin{cases} \tilde{u}_j(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)), & \text{if UAV } j \text{ reaches } d_j, \\ \tilde{u}_j(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)) + \gamma \max_{\mathbf{z}_j \in \mathcal{Z}_j} \mathbf{W}_{j,\text{out}}(\mathbf{z}_j(t+1), t+1) \\ \quad [\mathbf{v}'_j(t), \mathbf{x}'^{(1)}_j(t), \mathbf{x}'^{(2)}_j(t), \dots, \mathbf{x}'^{(n)}_j(t)], & \text{otherwise.} \end{cases} \quad (25)$$

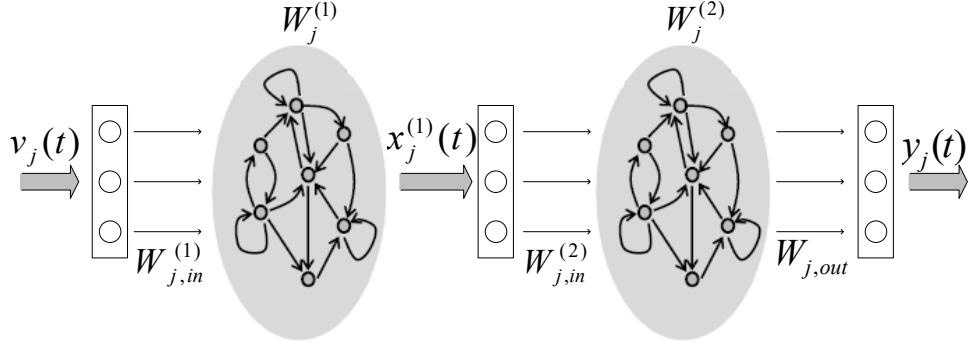


Fig. 1: Proposed Deep ESN architecture.

Here, $v'_j(t+1)$ and $\mathbf{x}'^{(n)}_j(t)$, correspond, respectively, to the next network state and reservoir state of layer (n) , at stage $(t+1)$, upon taking actions $\mathbf{z}_j(t)$ and $\mathbf{z}_{-j}(t)$ at stage t . Fig. 1 shows the proposed reservoir architecture of the deep ESN consisting of two layers.

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 [23] for updating the state transition functions $\mathbf{x}_j^{(n)}(t)$ at stage t . Therefore, the state transition function of the first layer $\mathbf{x}_j^{(1)}(t)$ will be:

$$\mathbf{x}_j^{(1)}(t) = (1 - \omega_j^{(1)})\mathbf{x}_j^{(1)}(t-1) + \omega_j^{(1)}\tanh(\mathbf{W}_{j,\text{in}}^{(1)}\mathbf{v}_j(t) + \mathbf{W}_j^{(1)}\mathbf{x}_j^{(1)}(t-1)), \quad (26)$$

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 , $\mathbf{x}_j^{(n)}(t)$, for $n > 1$ is given by:

$$\mathbf{x}_j^{(n)}(t) = (1 - \omega_j^{(n)})\mathbf{x}_j^{(n)}(t-1) + \omega_j^{(n)}\tanh(\mathbf{W}_{j,\text{in}}^{(n)}\mathbf{x}_j^{(n-1)}(t) + \mathbf{W}_j^{(n)}\mathbf{x}_j^{(n)}(t-1)), \quad (27)$$

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 $\mathbf{z}_j(t)$ and $\mathbf{z}_{-j}(t)$ of UAV j and other UAVs $(-j)$, respectively, for the current network state $\mathbf{v}_j(t)$ after training $\mathbf{W}_{j,\text{out}}$. It can be computed as:

$$y_j(\mathbf{v}_j(t), \mathbf{z}_j(t)) = \mathbf{W}_{j,\text{out}}(\mathbf{z}_j(t), t)[\mathbf{v}_j(t), \mathbf{x}_j^{(1)}(t), \mathbf{x}_j^{(2)}(t), \dots, \mathbf{x}_j^{(n)}(t)]. \quad (28)$$

We adopt a temporal difference RL approach for training the output matrix $\mathbf{W}_{j,\text{out}}$ 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 [24]:

Algorithm 1 Training phase of the proposed deep RL algorithm

Initialization:

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

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 $\mathbf{v}_j(t)$ based on (16).

Step 1: Action selection

 Each UAV j selects a random action $\mathbf{z}_j(t)$ with probability ϵ ,

 Otherwise, UAV j selects $\mathbf{z}_j(t) = \text{argmax}_{z_j \in \mathcal{Z}_j} y_j(\mathbf{v}_j(t), \mathbf{z}_j(t))$.

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 $\mathbf{z}_j(t)$.

Step 3: Reward computation

 Each UAV j computes its reward values based on (25).

Step 4: Action broadcast

 Each UAV j broadcasts its selected action $\mathbf{z}_j(t)$ to all other UAVs.

Step 5: Deep ESN update

- Each UAV j updates the state transition vector $\mathbf{x}_j^{(n)}(t)$ for each layer (n) of the deep ESN architecture based on (26) and (27).

- Each UAV j computes its output $y_j(\mathbf{v}_j(t), \mathbf{z}_j(t))$ based on (28).

- The weights of the output matrix $\mathbf{W}_{j,\text{out}}$ of each UAV j are updated based on the linear gradient descent update rule given in (29).

end for
end while
end for

$$\begin{aligned} \mathbf{W}_{j,\text{out}}(\mathbf{z}_j(t), t+1) = & \mathbf{W}_{j,\text{out}}(\mathbf{z}_j(t), t) + \lambda_j(r_j(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)) - y_j(\mathbf{v}_j(t), \mathbf{z}_j(t)))[\mathbf{v}_j(t), \\ & \mathbf{x}_j^{(1)}(t), \mathbf{x}_j^{(2)}(t), \dots, \mathbf{x}_j^{(n)}(t)]^T. \end{aligned} \quad (29)$$

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

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 $\mathbf{W}_{j,\text{out}}, \forall j \in \mathcal{J}$ have been optimized and is implemented on each UAV for execution during run time.

Algorithm 2 Testing phase of the proposed deep RL algorithm

```

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  $\mathbf{v}_j(t)$  based on (16).
    Step 1: Action selection
    Each UAV  $j$  selects an action  $\mathbf{z}_j(t) = \text{argmax}_{\mathbf{z}_j \in \mathcal{Z}_j} y_j(\mathbf{v}_j(t), \mathbf{z}_j(t))$ .
    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  $\mathbf{z}_j(t)$ .
    Step 3: Action broadcast
    Each UAV  $j$  broadcasts its selected action  $\mathbf{z}_j(t)$  to all other UAVs.
    Step 4: State transition vector update
    Each UAV  $j$  updates the state transition vector  $\mathbf{x}_j^{(n)}(t)$  for each layer  $(n)$  of the deep ESN architecture based on (26) and (27).
  end for
end while
  
```

During the training phase, each UAV aims at optimizing its output weight matrix $\mathbf{W}_{j,\text{out}}$ such that the value of the error function $e_j(\mathbf{v}_j(t))$ 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 (25). 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 [25]. 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 will be:

$$\pi_{j,z_j}(\mathbf{v}_j(t)) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{Z}_j|}, & \text{argmax}_{\mathbf{z}_j \in \mathcal{Z}_j} y_j(\mathbf{v}_j(t), \mathbf{z}_j(t)), \\ \frac{\epsilon}{|\mathcal{Z}_j|}, & \text{otherwise.} \end{cases} \quad (30)$$

Based on the selected action $\mathbf{z}_j(t)$, each UAV j updates its location, cell association, and transmission power level and computes its reward function according to (25). 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 $\mathbf{x}_j^{(n)}(t)$ for each layer (n) of the deep ESN architecture according to (26) and (27). The output y_j at stage t is then updated based on (28). Finally, the weights of the output matrix $\mathbf{W}_{j,\text{out}}$ of each UAV j are updated based on the linear gradient descent update rule given in (29). Note that, a UAV stops taking any actions once it has reached its destination. A summary of the training phase is given in Algorithm 1.

Meanwhile, the testing phase corresponds to the actual execution of the algorithm. In this phase, each UAV chooses its action greedily for each state $v_j(t)$, i.e., $\text{argmax}_{z_j \in \mathcal{Z}_j} y_j(v_j(t), z_j(t))$, and updates its location, cell association, and transmission power level accordingly. Each UAV then broadcasts its selected action and updates its state transition vector $x_j^{(n)}(t)$ for each layer n of the deep ESN architecture based on (26) and (27). A summary of the testing phase is given in Algorithm 2.

It is important to note that analytically guaranteeing the convergence of the proposed deep learning algorithm is challenging as it is highly dependent on the hyperparameters used during the training phase. For instance, using too few neurons in the hidden layers results in underfitting which could make it hard for the neural network to detect the signals in a complicated data set. On the other hand, using too many neurons in the hidden layers can either result in overfitting or an increase in the training time that could prevent the training of the neural network. Overfitting corresponds to the case when the model learns the random fluctuations and noise in the training data set to the extent that it negatively impacts the model's ability to generalize when fed with new data. Therefore, in this work, we limit our analysis of convergence by providing simulation results (see Section V) to show that, under a reasonable choice of the hyperparameters, convergence is observed for our proposed game. In such cases, it is important to study the convergence point and the convergence complexity of our proposed algorithm. Next, we characterize the convergence point of our proposed algorithm.

Proposition 1. If Algorithm 1 converges, then the convergence strategy profile corresponds to a SPNE of game \mathcal{G} .

Proof. An SPNE is a strategy profile that induces a Nash equilibrium on every subgame. Therefore, at the equilibrium state of each subgame, there is no incentive for any UAV to deviate after observing any history of joint actions. Moreover, given the fact that an ESN framework exhibits adaptive memory that enables it to store necessary previous state information, UAVs can essentially retain other players' actions at each stage t and thus take actions accordingly. To show that our proposed scheme guarantees convergence to an SPNE, we use the following lemma from [20].

Lemma 1. *For our proposed game \mathcal{G} , the payoff functions in (25) are bounded, and the number of players, state space and action space is finite. Therefore, \mathcal{G} is a finite game and hence a SPNE*

exists. This follows from Selten's theorem which states that every finite extensive form game with perfect recall possesses an SPNE where the players use behavioral strategies.

Here, it is important to note that for finite dynamic games of perfect information, any backward induction solution is a SPNE [18]. Therefore, given the fact that, for our proposed game \mathcal{G} , each UAV aims at maximizing its expected sum of *discounted rewards* at each stage t as given in (25), one can guarantee that the convergence strategy profile corresponds to a SPNE of game \mathcal{G} . This completes the proof. ■

Moreover, it is important to note that the convergence complexity of the proposed deep RL algorithm for reaching a SPNE is $O(J \times A^2)$. Next, we analyze the computational complexity of the proposed deep RL algorithm for practical scenarios in which the number of UAVs is relatively small.

Theorem 2. For practical network scenarios, the computational complexity of the proposed training deep RL algorithm is $O(A^3)$ and reduces to $O(A^2)$ when considering a fixed altitude for the UAVs, where A is the number of discretized unit areas.

Proof. Consider the case in which the UAVs can move with a fixed step size in a 3D space. For such scenarios, the state vector $\mathbf{v}'_j(t)$ is defined as:

$$\mathbf{v}'_j(t) = \left[\{\delta_{j,l,a}(t), \theta_{j,l,a}(t)\}_{l=1}^{L_j}, \theta_{j,d_j,a}(t), \{x_j(t), y_j(t), h_j(t)\}_{j \in \mathcal{J}} \right], \quad (31)$$

For each state $\mathbf{v}'_j(t)$, the action of UAV j is a function of the location, transmission power level and cell association vector of all other UAVs in the network. Nevertheless, the number of possible locations of other UAVs in the network is much larger than the possible number of transmission power levels and the size of the cell association vector of those UAVs. Therefore, by the law of large numbers, one can consider the number of possible locations of other UAVs only when analyzing the convergence complexity of the proposed training algorithm. Moreover, for practical scenarios, the total number of UAVs in a given area is considered to be relatively small as compared to the number of discretized unit areas i.e., $J \ll A$ (3GPP admission control policy for cellular-connected UAVs [1]). Therefore, by the law of large numbers and given the fact that the UAVs take actions in a parallel fashion, the computational complexity of our proposed algorithm is $O(A^3)$ when the UAVs update their x, y and z coordinates and reduces to $O(A^2)$ when considering fixed altitudes for the UAVs. This completes the proof. ■

Table I:
SYSTEM PARAMETERS

Parameters	Values	Parameters	Values
UAV max transmit power (\bar{P}_j)	20 dBm	SINR threshold ($\bar{\Gamma}_j$)	-3 dB
UE transmit power (\hat{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 (\hat{f})	2 GHz	Discount factor (γ)	0.7
# of hidden layers	2	Step size (\tilde{a}_j)	40 m
Leaky parameter/layer ($\omega_j^{(n)}$)	0.99, 0.99	ϵ	0.3

From Theorem 2, we can conclude that the convergence speed of the proposed training algorithm is significantly reduced when considering a fixed altitude for the UAVs. This in essence is due to the reduction of the state space dimension when updating the x and y coordinates only. It is important to note here that there exists a tradeoff between the computational complexity of the proposed training algorithm and the resulting network performance. In essence, updating the 3D coordinates of the UAVs at each step t allows the UAVs to better explore the space thus providing more opportunities for maximizing their corresponding utility functions. Therefore, from both Theorems 2 and 1, the UAVs can update their x and y coordinates only during the learning phase while operating within the upper and lower altitude bounds derived in Theorem 1.

V. SIMULATION RESULTS AND ANALYSIS

For our simulations, we consider an $800 \text{ m} \times 800 \text{ m}$ square area divided into $40 \text{ m} \times 40 \text{ m}$ 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 locations of the BSs and the ground UEs are randomized. The maximum transmit power for each UAV is discretized into 5 equally separated levels. We consider an uncorrelated Rician fading channel with parameter $\hat{K} = 1.59$ [26]. The external input of the deep ESN architecture, $v_j(t)$, 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 I summarizes the main simulation parameters.

Fig. 2a shows the upper bound for the optimal altitude of UAV j as a function of the SINR threshold value, $\bar{\Gamma}$, and for different transmit power levels, based on Theorem 1. On the other hand, Fig. 2b shows the lower bound for the optimal altitude of UAV j as a function of the

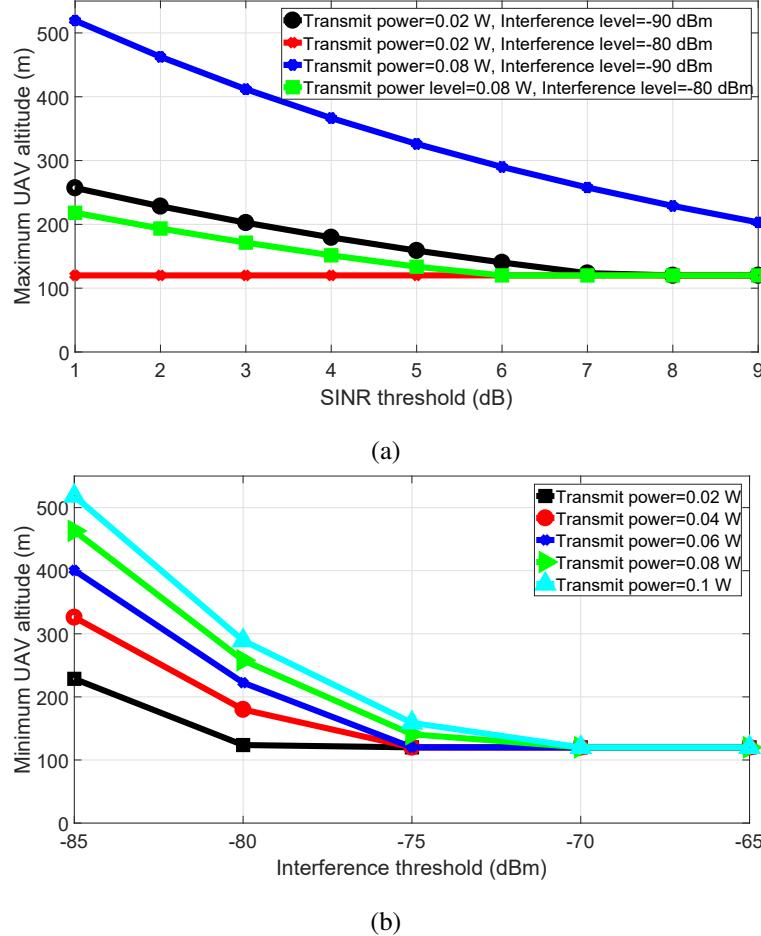


Fig. 2: The (a) upper bound for the optimal altitude of the UAVs as a function of the SINR threshold value ($\bar{\Gamma}$) and for different transmit power levels and ground network density and (b) lower bound for the optimal altitude of the UAVs as a function of the interference threshold value ($\sum_{c=1}^{C_{j,s}(t)} \bar{I}_{j,r,c,a}$) and for different transmit power levels.

SINR threshold value, $\bar{\Gamma}$, and for different transmit power levels and ground network density, based on Theorem 1. From Figs. 2a and 2b, we can deduce that the optimal altitude range of a given UAV is a function of network design parameters, ground network data requirements, the density of the ground network, and its action $v_j(t)$. For instance, the upper bound on the UAV's optimal altitude decreases as $\bar{\Gamma}$ increases while its lower bound decreases as $\sum_{c=1}^{C_{j,s}(t)} \bar{I}_{j,r,c,a}$ increases. Moreover, the maximum altitude of the UAV decreases as the ground network gets denser while its lower bound increases as the ground network data requirements increase. Thus, in such scenarios, a UAV should operate at higher altitudes. A UAV should also operate at higher altitudes when its transmit power level increases due to the increase in the lower and upper bounds of its optimal altitude.

Fig. 3 shows a snapshot of the path of a single UAV resulting from our approach and from

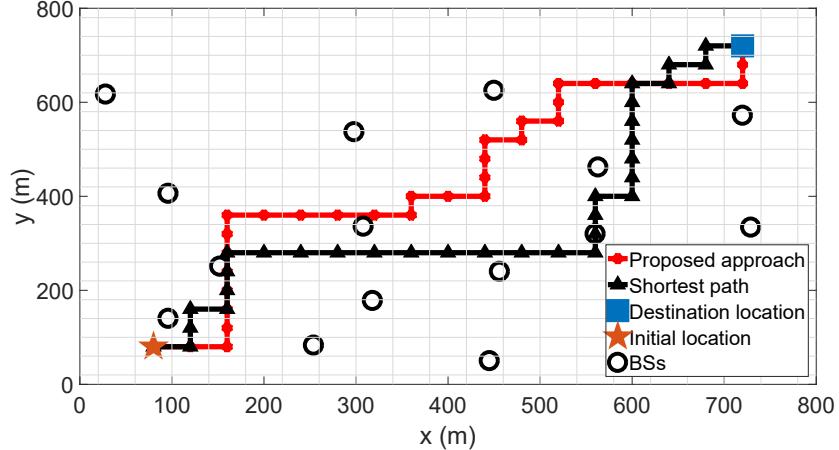


Fig. 3: Path of a UAV for our approach and shortest path scheme.

Table II:

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

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

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 reach their destinations with the minimum number of steps. Table II presents the performance results for the paths shown in Fig. 3. From Fig. 3, we can see that, for our proposed approach, the UAV selects a path away from the densely deployed area while maintaining proximity to its serving BS in a way that would minimize the steps required to reach its destination. This path will minimize the interference level that the UAV causes on the ground UEs and its wireless latency (Table II). From Table II, 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 that the UAV needs to reach the destination.

Fig. 4 compares the average values of the (a) wireless latency per UAV and (b) rate per ground UE resulting from our proposed approach and the baseline shortest path scheme. Moreover,

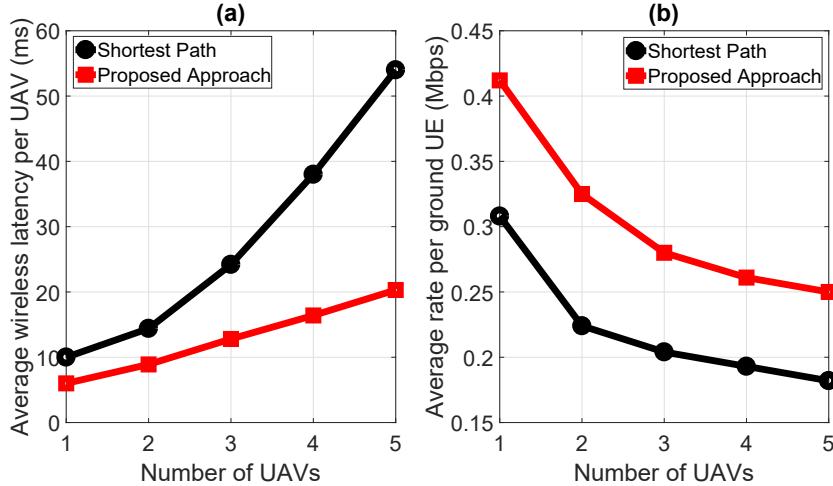


Fig. 4: Performance assessment of the proposed approach in terms of average (a) wireless latency per UAV and (b) rate per ground UE as compared to the shortest path approach, for different number of UAVs.

Table III compares the number of steps required by all UAVs to reach their corresponding destinations for the scenarios presented in Fig. 4. From Fig. 4 and Table III, we can see that, compared to the shortest path scheme, our approach achieves a lower wireless latency per UAV and a higher rate per ground UE for different numbers 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 ground UE data rate 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% increase in energy efficiency. Indeed, one can adjust the multi-objective weights of our utility function based on several parameters such as the rate requirements of the ground network, the power limitation of the UAVs, and the maximum tolerable wireless latency of the UAVs. Moreover, Fig. 4 shows that, as the number of UAVs increases, the average delay per UAV increases and the average rate per ground UE decreases, for all schemes. This is due to the increase in the interference level on the ground UEs and other UAVs as a result of the LoS link between the UAVs and the BSs.

Fig. 5 studies the effect of the UAVs' altitude on the average values of the (a) wireless latency per UAV and (b) rate per ground UE for different utility functions. From Fig. 5, we can see that, as the altitude of the UAVs increases, the average wireless latency per UAV increases for all studied utility functions. This is mainly due to the increase in the distance of the UAVs from their corresponding serving BSs which accentuates the path loss effect. Moreover, higher

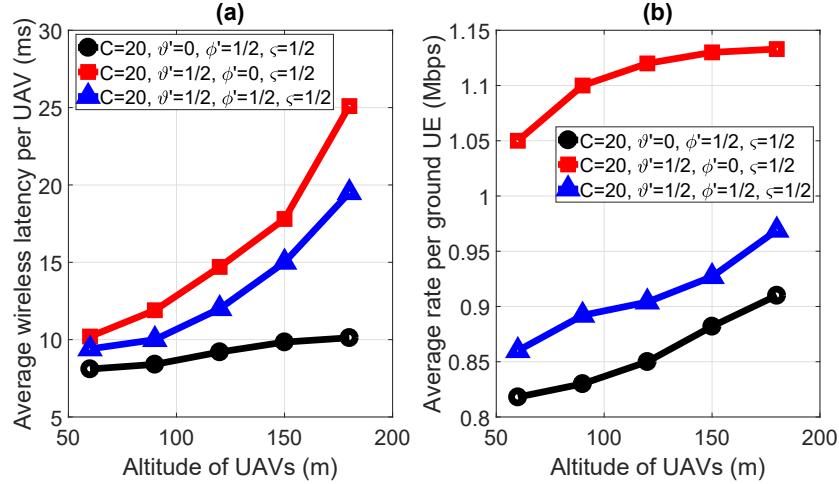


Fig. 5: Performance assessment of the proposed approach in terms of average (a) wireless latency per UAV and (b) rate per ground UE for different utility functions and for different altitudes of the UAVs.

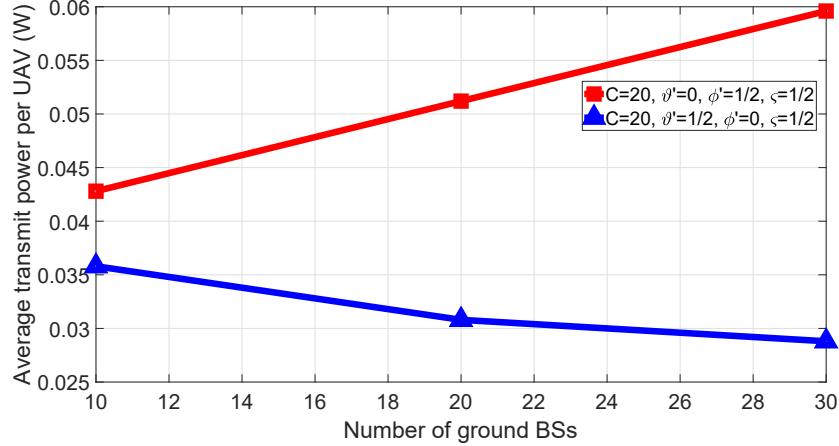


Fig. 6: Effect of the ground network densification on the average transmit power level of the UAVs along their paths.

UAV altitudes result in a higher average data rate per ground UE for all studied utility functions mainly due to the decrease in the interference level that is caused from the UAVs on neighboring BSs. Here, there exists a tradeoff between minimizing the average wireless delay per UAV and maximizing the average data rate per ground UE. Therefore, alongside the multiobjective weights, the altitude of the UAVs can be varied such that the ground UE rate requirements is met while minimizing the wireless latency for each UAV based on its mission objective.

Fig. 6 shows the average transmit power level per UAV along its path as a function of the number of BSs considering two utility functions, one for minimizing the average wireless latency for each UAV and the other for minimizing the interference level on the ground UEs. From Fig. 6, we can see that network densification has an impact on the transmission power level of

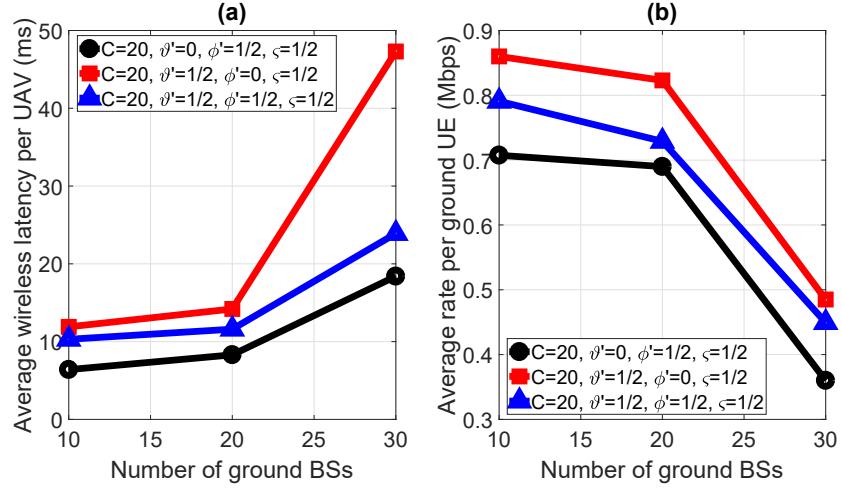


Fig. 7: Effect of the ground network densification on the average (a) wireless latency per UAV and (b) rate per ground UE for different utility functions and for a fixed altitude of 120m.

the UAVs. For instance, when minimizing the wireless latency of each UAV along its path, the average transmit power level per UAV increases from 0.04 W to 0.06 W as the number of ground BSs increases from 10 to 30, respectively. In essence, the increase in the transmit power level is the result of the increase in the interference level from the ground UEs as the ground network becomes denser. As a result, the UAVs will transmit using a larger transmission power level so as to minimize their wireless transmission delay. On the other hand, the average transmit power level per UAV decreases from 0.036 W to 0.029 W in the case of minimizing the interference level caused on neighboring BSs. This is due to the fact that as the number of BSs increases, the interference level caused by each UAV on the ground network increases thus requiring each UAV to decrease its transmit power level. Note that, when minimizing the wireless latency, the average transmit power per UAV is always larger than the case of minimizing the interference level, irrespective of the number of ground BSs. Therefore, the transmit power level of the UAVs is a function of their mission objective and the number of ground BSs.

Fig. 7 presents the (a) wireless latency per UAV and (b) rate per ground UE for different utilities as a function of the number of BSs and for a fixed altitude of 120 m. From this figure, we can see that, as the ground network becomes more dense, the average wireless latency per UAV increases and the average rate per ground UE decreases for all considered cases. For instance, when the objective is to minimize the interference level along with energy efficiency, the average wireless latency per UAV increases from 13 ms to 47 ms and the average rate per ground UE decreases from 0.86 Mbps to 0.48 Mbps as the number of BSs increases from 10 to

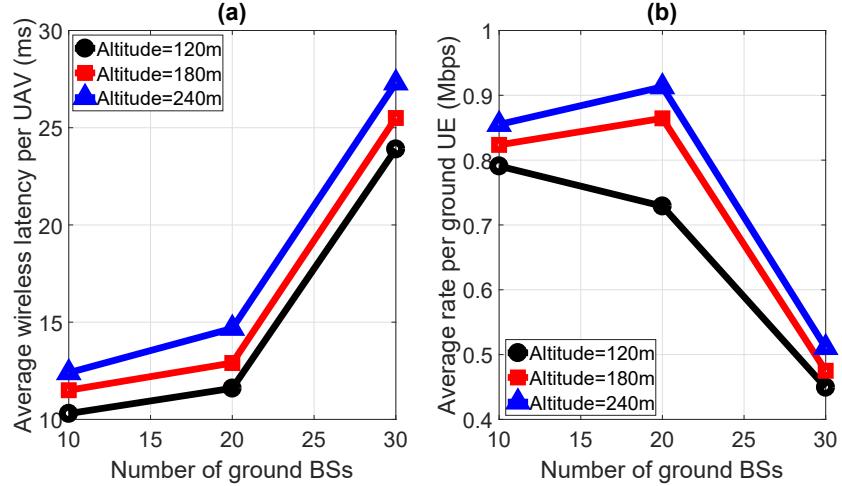


Fig. 8: Effect of the ground network densification on the average (a) wireless latency per UAV and (b) rate per ground UE for different utility functions and for various altitudes of the UAVs.

30. This is due to the fact that a denser network results in higher interference on the UAVs as well as other UEs in the network.

Fig. 8 investigates the (a) wireless latency per UAV and (b) rate per ground UE for different values of the UAVs altitude and as a function of the number of BSs. From this figure, we can see that as the UAV altitude increases and/or the ground network becomes denser, the average wireless latency per UAV increases. For instance, the delay increases by 27% as the altitude of the UAVs increases from 120 to 240 m for a network consisting of 20 BSs and increases by 120% as the number of BSs increases from 10 to 30 for a fixed altitude of 180 m. This essentially follows from Theorem 1 and the results in Fig. 2a which shows that the maximum altitude of the UAV decreases as the ground network gets denser and thus the UAVs should operate at a lower altitude when the number of BSs increases from 10 to 30. Moreover, the average rate per ground UE decreases as the ground network becomes denser due to the increase in the interference level and increases as the altitude of the UAVs increases. Therefore, the resulting network performance depends highly on both the UAVs altitude and the number of BSs in the network. For instance, in case of a dense ground network, the UAVs need to fly at a lower altitude for applications in which the wireless transmission latency is more critical and at a higher altitude in scenarios in which a minimum achievable data rate for the ground UEs is required.

Fig. 9 shows the effect of varying the number of nearest BSs (L_j) in the observed network state of UAV j , $v_j(t)$, on the average data rate per ground UE for different utility functions. From Fig. 9, we can see an improvement in the average rate per ground UE as the number

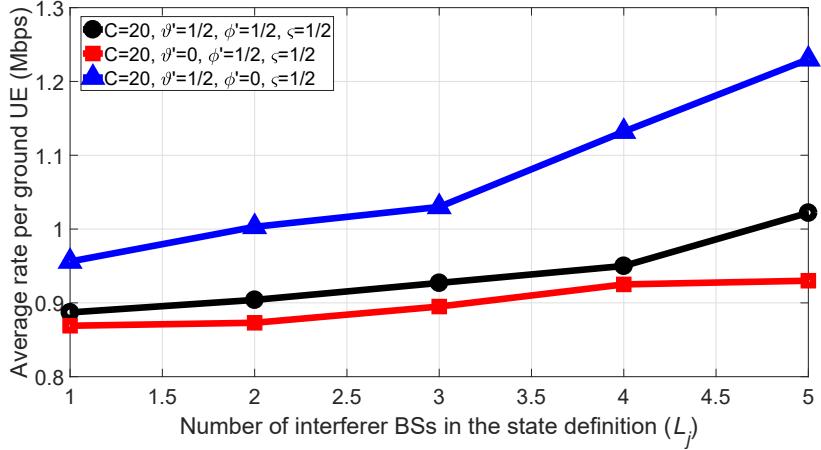


Fig. 9: The average rate per ground UE as a function of the number of interferer BSs in the state definition (L_j).

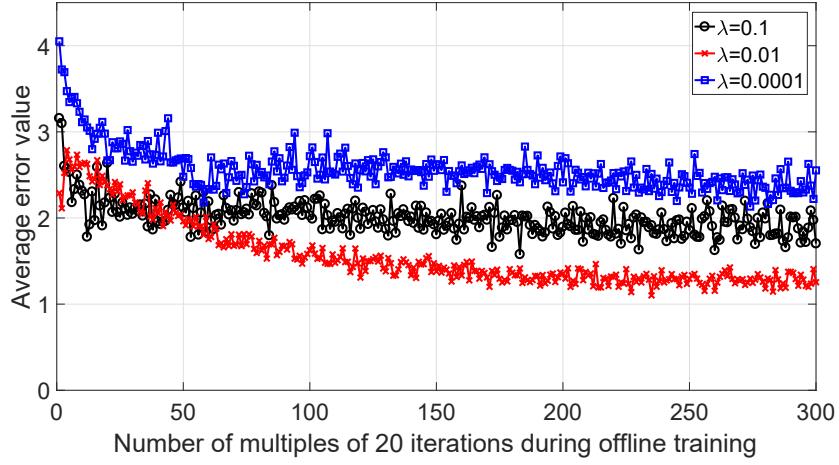


Fig. 10: Effect of the learning rate on the convergence of offline training.

of nearest BSs in the state definition increases. For instance, in scenarios in which the UAVs aim at minimizing the interference level they cause on the ground network along their paths, the average rate per ground UE increases by 28% as the number of BSs in the state definition increases from 1 to 5. This gain results from the fact that as L_j increases, the UAVs get a better sense of their surrounding environment and thus can better select their next location such that the interference level they cause on the ground network is minimized. It is important to note here, that as L_j increases, the size of the external input (v_j) increases thus requiring a larger number of neurons in each layer. This in turn increases the number of required iterations for convergence. Therefore, a tradeoff exists between improving the performance of the ground UEs and the running complexity of the proposed algorithm.

Fig. 10 shows the average of the error function $e_j(v_j(t))$ 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. 10, 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, $\lambda = 0.1$ 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 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 framework achieves better performance for $\lambda = 0.01$, as compared to smaller and larger values of the learning rate. We also note 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 [21].

VI. CONCLUSION

In this paper, we have proposed a novel interference-aware path planning scheme that allows cellular-connected UAVs to minimize the interference they cause on a ground network as well as their wireless transmission latency while transmitting online mission-related data. 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 it converges. The proposed algorithm enables each UAV to decide on its next location, transmission power level, and cell association vector in an autonomous manner thus adapting to the changes in the network. 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. The results have also shown that a UAV's altitude plays a vital role in minimizing the interference level on the ground UEs as well as the wireless transmission delay of the UAV. In particular, we have shown that the altitude of the UAV is a function of the ground network density, the UAV's objective and the actions of other UAVs in the network.

APPENDIX

A. Proof of Theorem 1

For a given network state $\mathbf{v}_j(t)$ and a particular action $\mathbf{z}_j(t)$, the upper bound for the altitude of UAV j can be derived when UAV j aims at minimizing its delay function only, i.e., $\vartheta' = 0$. For such scenarios, UAV j should guarantee an upper limit, $\bar{\Gamma}_j$, for the SINR value $\Gamma_{j,s,c,a}$ of the transmission link from UAV j to BS s on RB c at location a as given in constraint (13). Therefore, $\hat{h}_j^{\max}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))$ corresponds to the altitude at which UAV j achieves $\bar{\Gamma}_j$ and beyond which (13) is violated. The derivation of the expression of $\hat{h}_j^{\max}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))$ is:

$$\sum_{c=1}^{C_{j,s}(t)} \Gamma_{j,s,c,a} = \bar{\Gamma}_j, \quad (32)$$

$$\sum_{c=1}^{C_{j,s}(t)} \frac{\frac{\hat{P}_{j,s,a}(\mathbf{v}_j(t))}{C_{j,s}(t)} \cdot g_{j,s,c,a}(t)}{\left(\frac{4\pi\hat{f}d_{j,s,a}^{\max}}{\hat{c}}\right)^2 \cdot (I_{j,s,c}(t) + B_c N_0)} = \bar{\Gamma}_j, \quad (33)$$

$$\frac{\hat{P}_{j,s,a}(\mathbf{v}_j(t))}{C_{j,s}(t)} \cdot \frac{1}{\left(\frac{4\pi\hat{f}d_{j,s,a}^{\max}}{\hat{c}}\right)^2} \cdot \sum_{c=1}^{C_{j,s}(t)} \frac{g_{j,s,c,a}(t)}{I_{j,s,c}(t) + B_c N_0} = \bar{\Gamma}_j, \quad (34)$$

$$(d_{j,s,a}^{\max})^2 = \frac{\hat{P}_{j,s,a}(\mathbf{v}_j(t))}{C_{j,s}(t)} \cdot \frac{1}{\bar{\Gamma}_j \cdot \left(\frac{4\pi\hat{f}}{\hat{c}}\right)^2} \cdot \sum_{c=1}^{C_{j,s}(t)} \frac{g_{j,s,c,a}(t)}{I_{j,s,c}(t) + B_c N_0}, \quad (35)$$

where $d_{j,s,a}$ is the Euclidean distance between UAV j and its serving BS s at location a . Assume that the altitude of BS s is negligible, i.e., $z_s = 0$, $\hat{h}_j^{\max}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))$ can be expressed as:

$$\hat{h}_j^{\max}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)) = \sqrt{\frac{\hat{P}_{j,s,a}(\mathbf{v}_j(t))}{C_{j,s}(t) \cdot \bar{\Gamma}_j \cdot \left(\frac{4\pi\hat{f}}{\hat{c}}\right)^2} \cdot \sum_{c=1}^{C_{j,s}(t)} \frac{g_{j,s,c,a}(t)}{I_{j,s,c}(t) + B_c N_0} - (x_j - x_s)^2 - (y_j - y_s)^2}, \quad (36)$$

where x_s and y_s correspond to the x and y coordinates of the serving BS s and \hat{c} is the speed of light.

On the other hand, for a given network state $\mathbf{v}_j(t)$ and a particular action $\mathbf{z}_j(t)$, the lower bound for the altitude of UAV j can be derived when the objective function of UAV j is to minimize the interference level it causes on the ground network only, i.e., $\phi' = 0$ and $\varsigma = 0$. For such scenarios, the interference level that UAV j causes on neighboring BS r at location a should not exceed

a predefined value given by $\sum_{c=1}^{C_{j,s}(t)} \bar{I}_{j,r,c,a}$ ¹. Therefore, $\hat{h}_j^{\min}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))$ corresponds to the altitude at which UAV j achieves $\sum_{c=1}^{C_{j,s}(t)} \bar{I}_{j,r,c,a}$ and below which the level of interference it causes on BS r exceeds the value of $\sum_{c=1}^{C_{j,s}(t)} \bar{I}_{j,r,c,a}$. The derivation of the expression of $\hat{h}_j^{\min}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))$ is given by:

$$\sum_{c=1}^{C_{j,s}(t)} \sum_{r=1, r \neq s}^S \frac{\hat{P}_{j,s,a}(\mathbf{v}_j(t)) h_{j,r,c,a}(t)}{C_{j,s}(t)} = \sum_{c=1}^{C_{j,s}(t)} \sum_{r=1, r \neq s}^S \bar{I}_{j,r,c,a}, \quad (37)$$

$$\sum_{c=1}^{C_{j,s}(t)} \sum_{r=1, r \neq s}^S \frac{\hat{P}_{j,s,a}(\mathbf{v}_j(t)) \cdot g_{j,r,c,a}(t)}{C_{j,s}(t) \cdot \left(\frac{4\pi\hat{f}d_{j,r,a}^{\min}}{\hat{c}}\right)^2} = \sum_{c=1}^{C_{j,s}(t)} \sum_{r=1, r \neq s}^S \bar{I}_{j,r,c,a}, \quad (38)$$

To find $\hat{h}_j^{\min}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))$, we need to solve (38) for each neighboring BS r separately.

Therefore, for a particular neighboring BS r , (38) can be written as:

$$\sum_{c=1}^{C_{j,s}(t)} \frac{\hat{P}_{j,s,a}(\mathbf{v}_j(t)) \cdot g_{j,r,c,a}(t)}{C_{j,s}(t) \cdot \left(\frac{4\pi\hat{f}d_{j,r,a}^{\min}}{\hat{c}}\right)^2} = \sum_{c=1}^{C_{j,s}(t)} \bar{I}_{j,r,c,a}, \quad (39)$$

$$\frac{\hat{P}_{j,s,a}(\mathbf{v}_j(t)) \cdot \sum_{c=1}^{C_{j,s}(t)} g_{j,r,c,a}(t)}{C_{j,s}(t) \cdot \left(\frac{4\pi\hat{f}d_{j,r,a}^{\min}}{\hat{c}}\right)^2} = \sum_{c=1}^{C_{j,s}(t)} \bar{I}_{j,r,c,a}, \quad (40)$$

$$(d_{j,r,a}^{\min})^2 = \frac{\hat{P}_{j,s,a}(\mathbf{v}_j(t)) \cdot \sum_{c=1}^{C_{j,s}(t)} g_{j,r,c,a}(t)}{C_{j,s}(t) \cdot \left(\frac{4\pi\hat{f}}{\hat{c}}\right)^2 \cdot \sum_{c=1}^{C_{j,s}(t)} \bar{I}_{j,r,c,a}}, \quad (41)$$

where $d_{j,r,a}$ is the Euclidean distance between UAV j and its neighboring BS r at location a .

Assume that the altitude of BS r is negligible, i.e., $z_r = 0$, we have:

$$\hat{h}_{j,r}^{\min}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)) = \sqrt{\frac{\hat{P}_{j,s,a}(\mathbf{v}_j(t)) \cdot \sum_{c=1}^{C_{j,s}(t)} g_{j,r,c,a}(t)}{C_{j,s}(t) \cdot \left(\frac{4\pi\hat{f}}{\hat{c}}\right)^2 \cdot \sum_{c=1}^{C_{j,s}(t)} \bar{I}_{j,r,c,a}} - (x_j - x_r)^2 - (y_j - y_r)^2}, \quad (42)$$

Therefore, $\hat{h}_j^{\min}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))$ corresponds to the maximum value of $\hat{h}_{j,r}^{\min}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t))$ among all neighboring BSs r and is expressed as:

$$\hat{h}_j^{\min}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)) = \max_r \hat{h}_{j,r}^{\min}(\mathbf{v}_j(t), \mathbf{z}_j(t), \mathbf{z}_{-j}(t)), \quad (43)$$

where x_r and y_r correspond to the x and y coordinates of other neighboring BSs r . This completes the proof.

¹ $\sum_{c=1}^{C_{j,s}(t)} \bar{I}_{j,r,c,a}$ is a network design parameter that is a function of the ground network density, number of UAVs in the network and the data rate requirements of the ground UEs. The value of $\bar{I}_{j,r,c,a}$ is in fact part of the admission control policy which limits the number of UAVs in the network and their corresponding interference level on the ground network [1].

REFERENCES

- [1] 3GPP, "Enhanced lte support for aerial vehicles," Tech. Rep., Mar. 2017.
- [2] Qualcomm, "Paving the path to 5G: Optimizing commercial LTE networks for drone communication," Tech. Rep., Sep. 2016.
- [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. D. Muruganathan, S. Gao, H. Asplund, H. Maattanen, M. Bergstrom, S. Euler, and Y. . E. Wang, "The sky is not the limit: LTE for unmanned aerial vehicles," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 204–210, Apr. 2018.
- [5] M. M. Azari, F. Rosas, A. Chiumento, and S. Pollin, "Coexistence of terrestrial and aerial users in cellular networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2017, pp. 1–6.
- [6] T. Andre, K. A. Hummel, A. P. 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. IEEE Global Communications Conference*, Dec. 2017, pp. 1–6.
- [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. IEEE Wireless Communications and Networking Conference (WCNC)*, Mar. 2017, pp. 1–6.
- [9] M. Messous, S. Senouci, and H. Sedjelmaci, "Network connectivity and area coverage for UAV fleet mobility model with energy constraint," in *Proc. IEEE Wireless Communications and Networking Conference*, Apr. 2016, pp. 1–6.
- [10] M. M. Azari, F. Rosas, and S. Pollin, "Reshaping cellular networks for the sky: Major factors and feasibility," in *Proc. IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–7.
- [11] X. Wang, A. Chowdhery, and M. Chiang, "Networked drone cameras for sports streaming," in *Proc. IEEE International Conference on Distributed Computing Systems (ICDCS)*, Jun. 2017, pp. 308–318.
- [12] S. Zhang, Y. Zeng, and R. Zhang, "Cellular-enabled UAV communication: A connectivity-constrained trajectory optimization perspective," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2580–2604, Mar. 2019.
- [13] M. Bekhti, M. Abdennnebi, N. Achir, and K. Boussetta, "Path planning of unmanned aerial vehicles with terrestrial wireless network tracking," in *Proc. Wireless Days (WD)*, Mar. 2016, pp. 1–6.
- [14] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *Proc. IEEE Global Communications Conference*, Dec 2014, pp. 2898–2904.
- [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. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2007, pp. 82–87.
- [20] M. Osborne, *An Introduction to Game Theory*. Oxford University Press, 2004.
- [21] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks," *CoRR*, vol. abs/1710.02913, 2017.
- [22] C. Gallicchio and A. Micheli, "Echo state property of deep reservoir computing networks," *Cognitive Computation*, vol. 9, pp. 337–350, May 2017.
- [23] 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.
- [24] I. Szita and A. L. V. Gyenes, *Reinforcement Learning with Echo State Networks*. Springer, Berlin, Heidelberg, 2006, vol. 4131.
- [25] R. Sutton and A. Barto, *Introduction to Reinforcement Learning*. MIT Press, 1998.
- [26] A. Ghaffarkhah and Y. Mostofi, "Path planning for networked robotic surveillance," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3560–3575, Jul. 2012.