Optimizing Energy Efficiency in UAV-Assisted Networks Using Deep Reinforcement Learning

Babatunji Omoniwa, Boris Galkin, and Ivana Dusparic

***Abstract*—In this letter, we study the energy efficiency (EE) optimization of unmanned aerial vehicles (UAVs) providing wireless coverage to static and mobile ground users. Recent multi-agent reinforcement learning approaches optimise the system's EE using a 2D trajectory design, neglecting interference from nearby UAV cells. We aim to maximize the system's EE by jointly optimizing each UAV's 3D trajectory, number of connected users, and the energy consumed, while accounting for interference. Thus, we propose a cooperative Multi-Agent Decentralized Double Deep Q-Network (MAD-DDQN) approach. Our approach outperforms existing baselines in terms of EE by as much as 55 – 80%.**

***Index Terms*—Energy efficiency, UAV base stations, deep reinforcement learning, multi-agent system.**

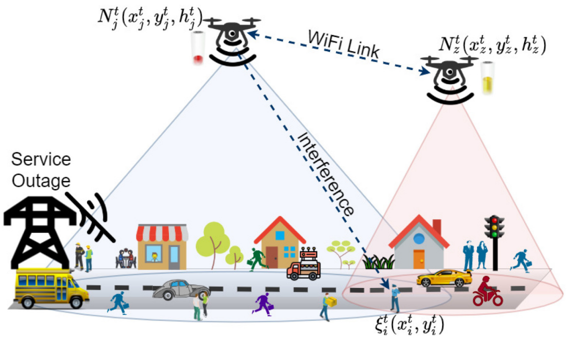


Fig. 1. System model for UAVs serving static and mobile ground users.

I. Introduction

THE DEPLOYMENT of unmanned aerial vehicles (UAVs) to provide wireless coverage to ground users has received significant research attention [1]–[7]. UAVs can play a vital role in supporting the Internet of Things (IoT) networks by providing connectivity to a large number of devices, static or mobile [1]. More importantly, UAVs have numerous real-world applications, ranging from assisted-communication in disaster-affected areas to surveillance, search and rescue operations [8], [9]. Specifically, UAVs can be deployed in circumstances of network congestion or downtime of existing terrestrial infrastructure. Nevertheless, to provide ubiquitous services to dynamic ground users, UAVs require robust strategies to optimise their flight trajectory while providing coverage. As energy-constrained UAVs operate in the sky, they may be faced with the challenge of interference from nearby UAV cells or other access points sharing the same frequency band, thereby impacting the system's energy efficiency (EE) [7].

There has been significant research effort on optimizing EE in multi-UAV networks [1]–[5]. The authors in [2] proposed an iterative algorithm to minimize the energy consumption of UAVs serving as aerial base stations to static ground users. In [4], a game-theoretic approach was proposed to maximize the system's EE while maximizing the ground area covered by the UAVs irrespective of the presence of ground users. However, these works rely on a central ground controller for UAVs' decision making, thereby making it impractical to be deployed for emergencies due to the significant amount of exchanged information between the UAVs and the controller. Moreover, it may be difficult to track user locations in such a scenario. Machine learning is increasingly being used to address complex multi-UAV deployment problems. In particular, multi-agent reinforcement learning (MARL) approaches have been deployed in several works to optimise the system's EE. A distributed Q-learning approach [1] focused on optimizing the energy utilisation of UAVs without considering the system's EE. To address this challenge, a deep reinforcement learning (DRL) approach [7] could be adopted. In our prior work [10], a DRL-based approach was proposed to optimise the EE of fixed-winged UAVs that move in circular orbits and are typically incapable of hovering like the rotary-winged UAVs. Moreover, the focus was on UAVs providing coverage to static ground users. The distributed DRL work in [3] was an improvement on the centralised approach in [5], where all UAVs are controlled by a single autonomous agent. The authors in [3], [5] proposed a deep deterministic policy gradient (DDPG) approach to improve the system's EE as UAVs hover at fixed altitudes while providing coverage to static ground users in an interference-free network environment. Although the approaches in [3] and [5] promise performance gains in terms of coverage score, they focus on the 2D trajectory optimization of the UAVs serving static ground users. Motivated by the research gaps above, we focus on maximizing the system's EE by optimizing the *3D trajectory* of each UAV over a series of time-steps, while taking into account the *impact of interference* from nearby UAV cells and the coverage of both *static and mobile ground users*. We propose a cooperative Multi-Agent Decentralized Double Deep Q-Network (MAD-DDQN) approach, where each agent's reward reflects the coverage performance in its neighbourhood. The

Manuscript received 22 March 2022; accepted 11 April 2022. Date of publication 14 April 2022; date of current version 9 August 2022. This work was supported in part by the Science Foundation Ireland (SFI) under Grant 16/SP/3804 (Enable) and Grant 13/RC/2077_P2 (CONNECT Phase 2), and in part by SFI and the National Natural Science Foundation of China (NSFC) through the SFI-NSFC Partnership Programme under Grant 17/NSFC/5224. The associate editor coordinating the review of this article and approving it for publication was K. Ota. *(Corresponding author: Babatunji Omoniwa.)*

The authors are with the CONNECT Centre, School of Computer Science and Statistics, Trinity College Dublin, Dublin 2, D02 PN40, Ireland (e-mail: omoniwab@tcd.ie; galkinb@tcd.ie; duspari@tcd.ie).

Digital Object Identifier 10.1109/LWC.2022.3167568

MAD-DDQN approach maximizes the system's EE without hampering performance gains in the network.

II. System Model

We consider a set of static and mobile ground users ξ located in a given area, as shown in Figure 1. Each user $i \in \xi$ at time *t* is located in the coordinate (x_i^t, y_i^t). We assume service unavailability from the existing terrestrial infrastructure due to disasters or increased network load. As such, a set *N* of quadrotor UAVs are deployed within the area to provide wireless coverage to the ground users. A serving UAV $j \in N$ at time *t* is located in the coordinate (x_j^t, y_j^t, h_j^t). Without loss of generality, we assume a guaranteed line-of-sight (LOS) channel condition [11], due to the aerial positions of the UAVs. Signal-to-interference-plus-noise-ratio (SINR) is a measure of the signal quality. It can be defined as the ratio of the power of a certain signal of interest and the interference power from all the other interfering signals plus the noise power. Each user $i \in \xi$ in time *t* can be connected to a single UAV $j \in N$ which provides the strongest downlink SINR. Thus, the SINR at time *t* is expressed as [1],

$$\gamma_{i,j}^t = \frac{\beta P(d_{i,j}^t)^{-\alpha}}{\Sigma_{z \in \chi_{int}} \beta P(d_{i,z}^t)^{-\alpha} + \sigma^2}, \tag{1}$$

where β and α are the attenuation factor and path loss exponent that characterises the wireless channel, respectively. σ^2 is the power of the additive white Gaussian noise at the receiver, $d_{i,j}^t$ is the distance between the *i* and *j* at time *t*. $\chi_{int} \in N$ is the set of interfering UAVs. *z* is the index of an interfering UAV in the set χ_{int}. *P* is the transmit power of the UAVs. We model the mobility of mobile users using the Gauss Markov Mobility (GMM) model [12], which allows users to dynamically change their positions. UAVs must optimise their flight trajectory to provide ubiquitous connectivity to users. Given a channel bandwidth B_w, the receiving data rate of a ground user can be expressed using Shannon's equation [7],

$$\mathbb{R}_{i,j}^t = B_w \log_2(1 + \gamma_{i,j}^t). \tag{2}$$

In our interference-limited system, coverage is affected by the SINR. Hence, we compute the connectivity score of a UAV $j \in N$ at time *t* as [3],

$$C_j^t = \sum_{\forall i \in \xi} w_j^t(i), \tag{3}$$

where $w_j^t(i) \in [0, 1]$ denotes whether user *i* is connected to UAV *j* at time *t*. $w_j^t(i) = 1$ if $\gamma_i^t = \gamma_{i,j}^t > \gamma_{th}$, otherwise $w_j^t(i) = 0$, where γ_{th} is the SINR predefined threshold. Likewise $\mathbb{R}_{i,j}^t = 0$ if user *i* is not connected to UAV *j*.

During flight operations, a UAV $j \in N$ at time *t* expends energy e_j^t. A UAVs' total energy consumption e_T is expressed as the sum in propulsion e_P and communication e_C energies, $e_T = e_P + e_C$. Since e_C is practically much smaller than e_P, i.e., $e_C \ll e_P$ [1], we ignore e_C. A closed-form analytical propulsion power consumption model for a rotary-wing UAV at time *t* is given as [13],

$$P(t) = \kappa_0 \left(1 + \frac{3V^2}{U_{tip}^2}\right) + \kappa_i \left(\sqrt{1 + \frac{V^4}{4v_0^4}} + \frac{V^2}{2v_0^2}\right)^{\frac{1}{2}} + \frac{\rho}{2}\nu s A V^3, \tag{4}$$

where κ_0 and κ_i are the UAVs' flight constants (e.g., rotor radius or weight), U_{tip} is the rotor blade's tip speed, v_0 is the mean hovering velocity, ν is the drag ratio, *s* is the rotor solidity, *A* is the rotor disc area, *V* is the UAVs' speed at time *t* and ρ is the air density. In particular, we take into account the basic operations of the UAV, such as hovering and acceleration. Therefore, we can derive the average propulsion power over all time-steps as $\frac{1}{T}\sum_{t=1}^{T} P(t)$, and the total energy consumed by UAV *j* at time *t* is given as [1],

$$e_j^t = \delta_t \cdot P(t), \tag{5}$$

where δ_t is the duration of each time-step. The EE of UAV *j* can be expressed as the ratio of the data throughput and the energy consumed in time-step *t*, expressed as,

$$\eta_j^t = \frac{\sum_{i \in \xi} \mathbb{R}_{i,j}^t}{e_j^t}. \tag{6}$$

III. Multi-Agent Reinforcement Learning Approach for Energy Efficiency Optimization

In this section, we formulate the problem and propose a MAD-DDQN algorithm to improve the trajectory of each UAV in a manner that maximizes the total system's EE.

A. Problem Formulation

Our objective is to maximize the total system's EE by jointly optimizing its 3D trajectory, number of connected users, and the energy consumed by the UAVs serving ground users under a strict energy budget. Maximizing the number of connected users C_j^t will maximize the total amount of data $\sum\limits_{i \in \xi} \mathbb{R}_{i,j}^t$ the UAV *j* will deliver in time-step *t* which, for a given amount of consumed energy e_j^t, will also maximize the total EE η_{tot}. Therefore, the optimization problem can be formulated as,

$$\max_{\forall j \in N:\ \mathrm{\mathbf{x}_j^t},\ \mathrm{\mathbf{y}_j^t},\ \mathrm{\mathbf{h}_j^t},\ \mathrm{\mathbf{e}_j^t},\ \mathrm{\mathbf{C}_j^t}} \eta_{tot} = \frac{\sum\limits_{t=1}^{T} \sum\limits_{j \in N} \sum\limits_{i \in \xi} \mathbb{R}_{i,j}^t}{\sum\limits_{t=1}^{T} \sum\limits_{j \in N} e_j^t} \tag{7a}$$

$$\text{s.t. } \gamma_{i,j}^t \geq \gamma_{th}, \quad \forall w_j^t(i) \in [0, 1], i, j, t, \tag{7b}$$

$$e_j^t \leq e_{\max}, \quad \forall j,\ t, \tag{7c}$$

$$x_{\min} \leq x_j^t \leq x_{\max}, \quad \forall j, t, \tag{7d}$$

$$y_{\min} \leq y_j^t \leq y_{\max}, \quad \forall j, t, \tag{7e}$$

$$h_{\min} \leq h_j^t \leq h_{\max}, \quad \forall j, t, \tag{7f}$$

where $e_{\max}$ is the maximum UAV energy level, x_{min}, y_{min}, h_{min} and x_{max}, y_{max}, h_{max} are the minimum and maximum 3D coordinates of *x*, *y* and *h*, respectively. As multiple wireless transmitters sharing the same frequency band are in close proximity to one another the possibility of interference is significantly increased. The computational complexity of problem (7a) is known to be NP-complete [6]. The problem (7a) is non-convex, thus having multiple local optimum. For this reason, solving (7a) with conventional optimization approaches is challenging [1], [6]. Specifically, the problem (7a) will become more complex as more UAVs are deployed in a shared wireless environment, hence it is challenging to find the optimal cooperative strategies to improve

Algorithm 1 Double Deep Q-Network (DDQN) for Agent *j*

```
1: Input: UAV3Dposition, ConnectivityScore, InstantaneousEnergyConsumed ∈ $S$ and Output: Q-values corresponding to each possible action $(+x_s, 0, 0)$, $(-x_s, 0, 0)$, $(0, +y_s, 0)$, $(0, -y_s, 0)$, $(0, 0, +z_s)$, $(0, 0, -z_s)$, $(0, 0, 0) \in A_j$.
2: $\mathcal{D}$ – empty replay memory, $\theta$ – initial network parameters,
   $\theta^-$ – copy of $\theta$, $\mathcal{N}_r$ – maximum size of replay memory,
   $\mathcal{N}_b$ – batch size, $\mathcal{N}^-$ – target replacement frequency.
3: $s \leftarrow$ initial state, maxStep $\leftarrow$ maximum number of steps in the episode
4: while goal not Reached and Agent alive and maxStep not reached do
5:     $s \leftarrow$ MapLocalObservationToState(Env)
6:         ▷ Execute $\epsilon$-greedy method based on $\pi_j$
7:     $a \leftarrow$ DeepQnetwork.SelectAction($s$)
8:         ▷ Agent executes action in state $s$
9:     $a$.execute(Env)
10:    if $a$.execute(Env) is True then
11:            ▷ Map sensed observations to new state $s'$
12:        Env.UAV3Dposition [6]
13:        Env.ConnectivityScore (3)
14:        Env.InstantaneousEnergyConsumed (5)
15:    $r \leftarrow$ Env.RewardWithCooperativeNeighbourFactor (8)
16:        ▷ Execute UpdateDDQNprocedure()
17:    Sample a minibatch of $\mathcal{N}_b$ tuples $(s, a, r, s') \sim Unif(\mathcal{D})$
18:    Construct target values, one for each of the $\mathcal{N}_b$ tuples:
19:    Define $a^{max}(s'; \theta) = \arg\max_{a'} Q(s', a'; \theta)$
20:    if $s'$ is Terminal then
21:        $y_j = r$
22:    else
23:        $y_j = r + \gamma Q(s', a^{max}((s'; \theta); \theta^-)$
24:    Apply a gradient descent step with loss $\parallel y_j - Q(s, a; \theta) \parallel^2$
25:    Replace target parameters $\theta^- \leftarrow \theta$ every $\mathcal{N}^-$ step
26: endwhile
```

the system's EE while completing the coverage tasks under dynamic settings. This is often because UAVs may become selfish and pursue the goal of improving their individual EE while minimizing the communication outage and energy consumption, rather than the collective goal of maximizing the system's EE. In such cases, cooperative MARL approaches may be suitable when individual and collective interests of UAVs conflict. Deep RL has been shown to perform well in decision-making tasks in such a dynamic environment [14]. Hence, we adopt a cooperative deep MARL approach to solve the system's EE optimization problem.

B. Cooperative Multi-Agent Decentralized Double Deep Q-Network (MAD-DDQN)

We propose a cooperative MAD-DDQN approach, where each agent's reward reflects the coverage performance in its neighbourhood. Here, each UAV is controlled by a Double Deep Q-Network (DDQN) agent that aims to maximize the system's EE by jointly optimizing its 3D trajectory, number of connected users, and the energy consumed. We assume the agents interact with each other in a shared and dynamic environment, which may lead to learning instabilities due to conflicting policies from other agents. From Algorithm 1, Agent *j* follows an ϵ–greedy policy by executing an action a, transiting from state s to a new state s' and receiving a reward reflecting the coverage performance in its neighbourhood in (8), after which DDQN procedure described on line 17–25 optimises the agent's decisions. We explicitly define the states, actions, and reward as follows:

- *State Space:* We consider the three-dimensional (3D) position of each UAV [6], the connectivity score and the UAV's instantaneous energy level at time *t*, expressed as a tuple, $\langle x^t : \{0, 1, \ldots, x_{max}\},\ y^t : \{0, 1, \ldots, y_{max}\},\ h^t : \{h_{min}, \ldots, h_{max}\},\ C_t, e_t\rangle$.
- *Action Space:* At each time-step $t \in T$, each UAV takes an action by changing its direction along the 3D coordinates. Unlike our closest related work and the evaluation baseline [3], we discretize the agent's actions following the design from [1] and [6], as follows: $(+x_s, 0, 0)$, $(-x_s, 0, 0)$, $(0, +y_s, 0)$, $(0, -y_s, 0)$, $(0, 0, +z_s)$, $(0, 0, -z_s)$ and (0, 0, 0). Our rationale to discretize the action space was to ensure quick adaptability and convergence of the agents.
- *Reward:* The agent's goal is to learn a policy that implicitly maximizes the system's EE by jointly minimizing the ground users outage and total UAVs energy consumption. Hence, we introduce a shared cooperative factor $\mho$ to shape the reward formulation of each agent *j* in each time-step $t \in T$ given as,

$$\mathcal{R}_j^t = \begin{cases} \mho + \omega + 1, & \text{if } C_j^t > C_j^{t-1} \\ \mho + \omega, & \text{if } C_j^t = C_j^{t-1} \\ \mho + \omega - 1, & \text{otherwise,} \end{cases} \tag{8}$$

where C_j^t and C_j^{t-1} are the connectivity score in present and previous time-step, respectively. $\omega = \frac{e_j^{t-1} - e_j^t}{e_j^t + e_j^{t-1}}$, where e_j^t and e_j^{t-1} are the instantaneous energy consumed by agent *j* in present and previous time-step, respectively. To enhance cooperation, we assign each agent a '$+1$' incentive from its neighbourhood via a function $\mho$ only when the overall connectivity score, which is the total number of connected users by UAVs in its locality in the present time-step C_t^o exceeds that in the previous time-step C_{t-1}^o, otherwise the agent receives a '-1' incentive. We compute $\mho$ as,

$$\mho = \begin{cases} +1, & \text{if } C_t^o > C_{t-1}^o \\ -1, & \text{otherwise.} \end{cases} \tag{9}$$

C. DDQN Implementation

The neural network (NN) architecture of Agent *j*'s DDQN shown in Figure 2 comprises of a 5-dimensional state space input vector, densely connected to 2 layers with 128 and 64 nodes, with each using a rectified linear unit (ReLU) activation function, leading to an output layer with 7 dimensions. Our decentralized approach assume agents to be independent learners. Following the analysis presented in [15], the computational complexity of the NN architecture used in the MAD-DDQN is approximately $\mathcal{O}(D_s KW)$ with an average response time of 5.6 ms, while that of our closest related work and the evaluation baseline [3] (MADDPG) is approximately $\mathcal{O}(D_s KW) + \mathcal{O}((D_a + D_s)KW)$ with an average response time of 7.4 ms, where D_s is the dimension of the state space, D_a is the dimension of the action space, K is the number of layers, and W is the number nodes in each hidden layer.

In the training phase, given the state information as input, Agent *j* trains the main network to make better decisions by yielding Q-values corresponding to each possible action as output. The maximum Q-value obtained determines the action the agent executes. At each time-step Agent *j* observes its present state s and updates it's trajectory by selecting an action a in accordance with its policy. Following its action in time-step *t*,

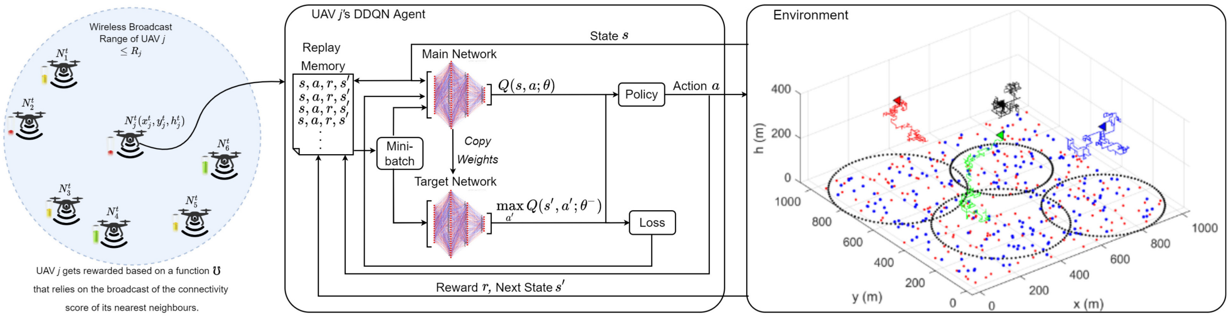


Fig. 2. Multi-agent decentralized double deep Q-network framework where each UAV *j* equipped with a DDQN agent interacts with its environment. The environment shows the simulation snapshot of UAVs providing wireless coverage to 200 static (blue) and 200 mobile (red) ground users with flight trajectories. On the left shows the broadcast range of UAV *j* in a multi-UAV scenario, where UAVs broadcast their telemetry information to nearest neighbours.

Agent *j* observes a reward r which is defined in (8), and transits to a new state s'. The information (s, a, r, s') is inputted in the replay memory as shown in Figure 2. Agent *j* then samples the random mini-batch from the replay memory and uses the mini-batch to obtain y_j. The optimization is performed with $L(\theta)$ and θ updated accordingly. In every 100th time-step, the target Q-network updates the parameters θ^- with the same parameters θ of the main network. For the training, the memory size was set to 10,000, and the mini-batch size was set to 1024. The optimization is performed using a variant of the stochastic gradient descent called RMSprop to minimize the loss following the methodology described in [16, Ch. 4]. The learning rate and discount factor were set to 0.0001 and 0.95, respectively. We train the Q-networks by running multiple episodes, and at each training step the ϵ-greedy policy is used to have a balance between exploration and exploitation [16]. In the ϵ-greedy policy, the action is randomly selected with ϵ probability, whereas the action with the largest action value is selected with a probability of $1 - \epsilon$. The initial value of ϵ was set to 1 and linearly decreased to 0.01.

IV. Evaluation and Results

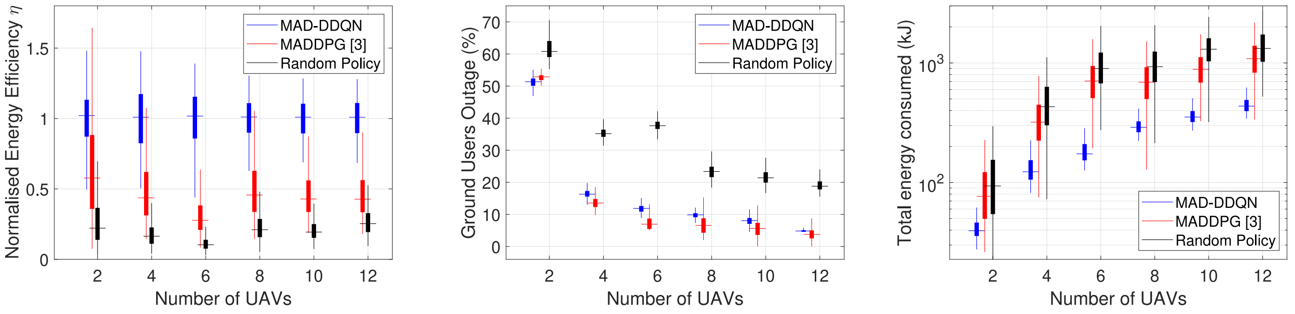
In this section, we verify the effectiveness of the proposed MAD-DDQN approach against the following baselines: (*i*) the random policy and (*ii*) the MADDPG [3] approach that considers a 2D trajectory optimization while neglecting interference from nearby UAV cells. Simulation parameters are presented in Table I. We simulate a varying number of UAVs ranging from 2 to 12 to serve both static and mobile ground users in a 1000 × 1000 m^2 area as shown in Figure 2. We perform 2000 runs of Monte-Carlo (MC) trials over trained episodes. In Figure 3, we compare the MAD-DDQN approach with baselines to evaluate the impact of different number of deployed UAVs on the EE, ground users outage and total energy consumption. Due to baseline MADDPG approach taking significantly longer to converge (learn suitable behaviors), to achieve a fair comparison, Figure 3 compares the performance after training the MAD-DDQN approach for 250 episodes and the MADDPG approach for 2000 episodes.

Since we focus on comparing the EE values rather than showing their absolute values, we normalise the EE values with respect to the mean values of the proposed MAD-DDQN approach. From Figure 3(a), we observe that the MAD-DDQN approach consistently outperforms the random policy and MADDPG approaches across the entire range of UAVs deployment by approximately 80% and 55%, respectively.

TABLE I
Simulation Parameters

Parameters	*Value*
Software platform/Library	Python 3.7.4/PyTorch 1.8.1
Optimiser/Loss function	RMSprop/MSELoss
Learning rate/Discount factor	0.0001/0.95
Hidden layers/Activation function	2 (128, 64)/ReLu
Replay memory size/Batch size	10,000/1024
Policy/Episodes/maxStep	ϵ-greedy/250/1500
No. of ground users/Model	400/GMM
Ground user direction/Velocity	[0, 2π]/[0, 15] m/s
Number of UAVs/Weight per UAV	[2–12]/16 kg
Nominal battery capacity	16,000 mAh
Maximum transmit power [6]	20 dBm
Noise power/SINR threshold [2]	-130 dBm/5 dB
Bandwidth [6]	1 MHz
Pathloss exponent [2], [6]	2
UAV step distance ($\forall\ x_s, y_s, z_s$)	[0–20] m

Interestingly, we see a marginally better performance by the MADDPG approach over the MAD-DDQN approach in minimizing the outages experienced by ground users by about 2%, as shown in Figure 3(b). However, the slight performance gain by the MADDPG comes at a huge computational training cost which is 8 times higher than the MAD-DDQN approach. Intuitively, the MAD-DDQN approach hides redundant information about the environment through discretization of the agent's action space, which makes the MAD-DDQN approach require less experience to successfully learn a policy than the MADDPG approach. On the other hand, the random policy performed worst among the approaches in reducing connection outages, emphasizing the relevance of strategic decision making in MARL problems. Figure 3(c) clearly shows that the proposed approach significantly minimizes the total energy consumed by all UAVs as compared to the baselines. Although the MADDPG approach performs slightly better at reducing outages than our approach, our MAD-DDQN approach is significantly more energy efficient, thereby implying the MADDPG approach trades energy consumption for improved coverage of ground users. In Figure 4, we show the plot of the EE versus the learning episodes while varying the number of agents to demonstrate the convergence behavior of the MAD-DDQN approach. We observe a steady decrease in the converged values of the EE while increasing the number of UAVs because the system becomes more unstable with more UAVs, thereby decreasing the system throughput as interference increases. Overall, the cooperative MAD-DDQN approach shows convergence in the system's EE irrespective of the number of UAVs deployed in the network.



(a) Energy efficiency η vs. number of UAVs.

(b) Ground users outage vs. number of UAVs.

(c) Total energy consumed vs. number of UAVs.

Fig. 3. Impact of number of deployed UAVs on the UAVs' EE, ground users outage and total energy consumption under dynamic network conditions with 400 ground users deployed in a 1 km^2 area, with results from 2000 runs of MC trials.

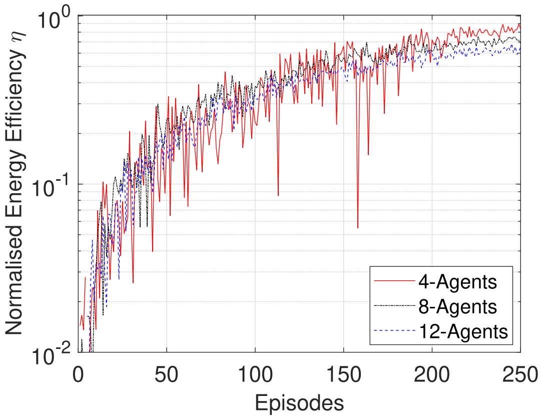


Fig. 4. Energy efficiency η vs. learning episodes showing the convergence of MAD-DDQN while varying the number of agents.

V. Conclusion

In this letter, we propose a MAD-DDQN approach to optimise the EE of a fleet of UAVs serving static and mobile ground users in an interference-limited environment. The MAD-DDQN approach guarantees quick adaptability and convergence, thereby allowing agents to learn policies that maximize the total system's EE by jointly optimizing its 3D trajectory, number of connected users, and the energy consumed by the UAVs serving ground users under a strict energy budget. Extensive simulation results have demonstrated that the MAD-DDQN approach significantly outperforms the random policy and a state-of-the-art decentralized MARL solution in terms of EE without degrading coverage performance in the network.

References

[1] B. Omoniwa, B. Galkin, and I. Dusparic, "Energy-aware optimization of UAV base stations placement via decentralized multi-agent Q-learning," in *Proc. IEEE 19th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2022, pp. 216–222.

[2] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient Internet of Things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, Nov. 2017.

[3] C. H. Liu, X. Ma, X. Gao, and J. Tang, "Distributed energy-efficient multi-UAV navigation for long-term communication coverage by deep reinforcement learning," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1274–1285, Jun. 2020.

[4] L. Ruan *et al.*, "Energy-efficient multi-UAV coverage deployment in UAV networks: A game-theoretic framework," *China Commun.*, vol. 15, no. 10, pp. 194–209, Oct. 2018.

[5] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.

[6] X. Liu, Y. Liu, and Y. Chen, "Reinforcement learning in multiple-UAV networks: Deployment and movement design," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8036–8049, Aug. 2019.

[7] B. Galkin, E. Fonseca, R. Amer, L. A. DaSilva, and I. Dusparic, "REQIBA: Regression and deep Q-learning for intelligent UAV cellular user to base station association," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 5–20, Jan. 2022.

[8] C. Zhang, M. Dong, and K. Ota, "Heterogeneous mobile networking for lightweight UAV assisted emergency communication," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 3, pp. 1345–1356, Sep. 2021.

[9] J. Xu, K. Ota, and M. Dong, "Big data on the fly: UAV-mounted mobile edge computing for disaster management," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2620–2630, Oct.–Dec. 2020.

[10] B. Galkin, B. Omoniwa, and I. Dusparic, "Multi-agent deep reinforcement learning for optimising energy efficiency of fixed-wing UAV cellular access points," in *Proc. IEEE Int. Conf. Commun.*, May 2022, pp. 1–6.

[11] B. Galkin, J. Kibilda, and L. A. DaSilva, "Deployment of UAV-mounted access points according to spatial user locations in two-tier cellular networks," in *Proc. Wireless Days (WD)*, 2016, pp. 1–6.

[12] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. Mobile Comput.*, vol. 2, no. 5, pp. 483–502, 2002.

[13] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.

[14] M. Zhang, S. Fu, and Q. Fan, "Joint 3D deployment and power allocation for UAV-BS: A deep reinforcement learning approach," *IEEE Wireless Commun. Lett.*, vol. 10, no. 10, pp. 2309–2312, Oct. 2021.

[15] J. Hribar, A. Marinescu, A. Chiumento, and L. A. DaSilva, "Energy aware deep reinforcement learning scheduling for sensors correlated in time and space," *IEEE Internet Things J.*, early access, Sep. 21, 2021, doi: 10.1109/JIOT.2021.3114102.

[16] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *Found. Trends Mach. Learn.*, vol. 11, nos. 3–4, p. 156, 2018.