Autonomous Flood Area Coverage using Decentralized Multi-UAV System with Directed Explorations

Armaan Garg and Shashi Shekhar Jha
Indian Institute of Technology Ropar
India
(armaan.19csz0002,shashi)@iitrpr.ac.in

ABSTRACT

During floods, access to real-time ground information is of critical importance for disaster response teams. Unmanned Aerial Vehicles (UAVs) can be quickly deployed in such disaster scenarios for gauging the ground situation. In this paper, we present a decentralized multi-UAV control algorithm based on deep reinforcement learning for flood area coverage. The UAVs are tasked to access the risk levels of the flooded regions and autonomously distribute themselves in order to gather ground information from flood areas in a time-sensitive manner. The task is time-sensitive due to the limited battery of the UAVs and the human lives at risk. In our proposed approach, we follow the paradigm of decentralized training and decentralized execution with opportunistic communication wherein each UAV makes individual decisions based on the information captured locally and the information received via intermittent communication with other UAVs. Further, to learn the best-performing control policy, a flood-water flow estimation algorithm called D8 is employed. With D8, we utilize the domain knowledge to generate better exploration strategies for boosting the initial policy gradients in the right direction. Experiments are performed over real-world inspired simulated flood environments. The proposed decentralized multi-UAV control model, dec-DQNC8, is compared with other prevalent techniques from the literature. The results highlight the significance of the proposed model as it outperforms other techniques and moreover, has the optimal performance when evaluated over a test environment.

CCS CONCEPTS

• **Computing methodologies** → **Multi-agent systems**; *Multi-agent planning*; **Reinforcement learning**.

KEYWORDS

Unmanned Aerial Vehicles (UAVs), Deep Reinforcement Learning, Real-Time Coverage, Path Planning, Disaster management

1 INTRODUCTION

Access to ground information during any form of natural disaster is of paramount importance for planning rescue and response missions. In recent times, Unmanned Aerial Vehicles (UAVs) have become a potential platform to gather ground information by performing area coverage. UAVs' quick deployment and multi-functional capabilities, such as reconnaissance, aerial data imaging, remote sensing, tracking and coverage, aerial delivery, etc. [29], have led to their use in myriad of real-world applications. However, there has been limited use of multi-UAV systems for autonomous area coverage applications during natural disasters. In this paper, we consider the decentralized deployment of multiple UAVs having a limited battery life for gathering ground information during a flood disaster. Deploying a decentralized multi-UAV system is non-trivial due to the challenges related to the stochasticity of the environment, autonomous handling of the UAVs trajectories, lack of information sharing and communication, and the limited energy of UAVs among others [5, 6, 25].

The decentralized multi-UAV based area coverage can be formulated as a decentralized partially-observable MDP (dec-POMDP) [26]. In literature, various researchers have attempted to solve the dec-POMDP using the model-free multi-agent reinforcement learning (MARL) approaches [9]. RL algorithms [30] address the problem of learning autonomous controls by interacting with the environment in a trial-and-error fashion. However, simple RL techniques (usually tabular methods such as Q-Learning), have limited applicability when it comes to large state space problems. To learn control policies in environments with huge state spaces, Deep Reinforcement Learning (DRL) [24] methods have shown good performances in recent times. DRL uses deep neural network based non-linear function approximators to capture the complexities of the state-space without explicitly specifying the state features. DRL has been used in a multitude of works to learn control policies for autonomous UAVs to perform the desired task(s). In [6], authors discuss a detailed review of DRL methods and their applicability for learning autonomous control of UAVs without any human intervention. In [11], we presented a centralized algorithm D8DQN for a multi-UAV area coverage problem highlighting the impact of domain knowledge on RL policy learning.

As discussed, the majority of the prominent work done in the field of multi-agent reinforcement learning (MARL) considers some form of centralized entity to make individual agents learn from global knowledge of the environment [8, 13, 17, 20]. The global information is captured from each agent and stored in a centralized unit. However, in real-world settings, deploying a centralized system to train a multi-UAV policy during natural disasters is not practical as usually the data captured by each UAV is distributed over the environment. Further, the location of the centralized entity should be known by the UAVs at all times and they need to have bi-directional communication with the central server irrespective of their distances. This is usually impractical to achieve where the environment is large, dynamic and stochastic [10].

Hence, various researchers have attempted to solve dec-POMDP problems using decentralized DRL approaches [18, 19, 22, 33]. However, the problem of learning a decentralized multi-UAV policy using DRL approaches has received very limited attention in the

Proc. of the Adaptive and Learning Agents Workshop (ALA 2023), Cruz, Hayes, Wang, Yates (eds.), May 29-30, 2023, London, UK, https://alaworkshop2023.github.io/. 2023.

literature. In this paper, we propose a DRL based decentralized multi-UAV system to operate in a highly dynamic flood environment for performing autonomous area coverage tasks. The objective here is to gather critical ground information of a flooded area using multiple autonomous UAVs with limited energy for relief and evacuation purposes. We further employ a domain-knowledge based estimator to provide directed explorations to the multi-UAV system during the initial phase of policy learning. In addition, we consider limited communication among the UAVs to exchange their experience in order to improve the local UAV policies. This helps the UAVs to learn about other UAVs and their observation histories and motivates them to have segregated trajectories.

The key contribution of this paper include:

(1) A decentralized multi-UAV control policy utilizing domain knowledge to perform autonomous area coverage tasks during floods.
(2) Sharing experiences among UAVs using opportunistic communication based on their current energy and inter-UAV separation.
(3) Introducing coverage maps to generate non-overlapping UAV trajectories to maximize the coverage of unobserved locations.
(4) Real-world dynamics, such as surface elevation levels and flood water rise magnitude are incorporated into the simulated environment for better practical sense.

2 RELATED WORK

In related literature two major techniques are highlighted to achieve cooperation among the UAVs in a multi-agent setting, one being the use of a centralized entity [25, 28, 35] and the other being partially or fully decentralized systems [18, 19, 22, 33]. For any form of cooperation, it's important to have some form of communication such that the agents are aware of the state and/or actions of other agents so as to perform the most appropriate action that leads to the highest global reward. Many previous studies have attempted to learn multi-UAV policies using a centralized entity so as to make the UAVs aware of the global state and perform the desired task(s). As in [25], the authors propose a mathematical optimisation model based on particle swarm optimisation (PSO) to effectively capture maximum images of the impacted region. This study considers a known environment and assumes centralized association among multiple UAVs to address the maximum area coverage challenge in a cooperative manner. Another study on area coverage is discussed in [35], where the authors assumed that a leader UAV has the knowledge of the global state and other UAVs in the system can communicate with its neighbour. After each UAV is aware of the global state, cooperative actions are executed. This task was formulated as a quadratic programming problem and was solved using the Sequential Quadratic Programming (SQP) algorithm. In [31], the authors proposed a distributed PSO model to perform exploration of a disaster area using UAVs. There is no central node considered in this distributed approach, but the UAVs are able to share their local information with other UAVs that are in their vicinity.

However, it's difficult to achieve desirable policies with relatively simpler techniques such as iterative methods (SQP) and heuristic models (PSO) especially when the environment is dynamic. Also, centralized systems are difficult to deploy as the agents are usually distributed over the environment and sometimes it's infeasible to keep all the agents connected to a centralized entity via a bi-direction communication link. Recent studies have proposed decentralized RL systems to cope with the impracticality of centralized entities to learn multi-UAVs controls via interacting with the environment in a trial-and-error fashion. In [7], authors employed a DRL technique known as Deep Q-Networks (DQN) for trajectory planning of multiple UAVs for flood monitoring tasks. It was assumed that UAVs have infinite battery life and a UAV is able to gather information related to the heading angle and bank angle of other UAVs, however, no communication protocol was employed. In another recent study [36], authors discussed the idea of multi-UAV based content coverage for ground users. The constraint of limited battery and caching storage of the UAVs is also addressed in [36] along with the coupled trajectory planning of the UAVs. A decentralized energy-efficient multi-UAV trajectory planning algorithm is proposed using the Q-learning approach. In [34], authors presented an overview of theories and algorithms for MARL models, highlighting the theoretical results of MARL algorithms with the types of tasks they address, i.e., fully cooperative, fully competitive, and a mix of the two. Further, the taxonomies of the MARL theory were discussed w.r.t. decentralized systems with networked agents, the convergence of policy-based methods and learning in extensive-form games.

However, none of these works make use of domain knowledge to learn the RL policy which we address in this paper. We provide a DRL specific solution to learn multi-UAV controls tasked to perform area coverage of flooded regions. Constraints regarding UAVs' limited energy and range specific communication among the UAVs are other key components of this paper.

3 PRELIMINARIES

In this section, we discuss the flood environment along with its features. Furthermore, we present the underlying idea behind the working of the D8 algorithm that utilizes the domain knowledge to generate action estimates for UAVs and provides an improved exploration strategy.

3.1 Environment Description

Here, we formally describe the flood environment used for modelling the multi-UAV area coverage. The environment is considered as a 2D terrain divided into $n \times m$ number of cells of equal size. The dimension of a cell is equal to the *Field of View (FoV)* of the UAV. The FoV of a UAV is a rectangular area captured by the UAV's ventral camera and its dimension depends on the UAV's altitude and camera angles. Real-world elevation information of the terrain's surface is collected using the Topographic map tool [4] and is incorporated while rendering the environment.

To simulate flood, a 2D water mask is overlayed over the terrain layer. The dynamics of flood water are defined using two parameters, namely, water flow rate f_{rate} and water level w_l. Note that the change in w_l as detected from the images captured from the FoV of the UAV is negligible corresponding to the altitude of the UAV. Each cell in the environment also contains information on the

human population density that is mapped using the Flood Mapping tool [1]. We define a risk level $\mathcal{U}_l^c(.,.)$ for a cell c as:

$$\mathcal{U}_l^c(w_l^c, p_l^c) = w_l^c \times p_l^c \tag{1}$$

where, p_l^c is the human population density at cell c and w_l^c represents its current water level. Both, w_l^c and p_l^c have a discrete and finite number of levels and so does $\mathcal{U}_l^c$.

3.2 The D8 Flow Estimation

The D8 technique [15] generates water flow directions based on the estimation of water discharge directions. D8 estimates the cell with the largest water accumulation in the cell's neighbourhood under a UAV's observation. It utilizes the surface elevation information corresponding to the cells to provide a flow direction estimate. In the considered scenario, the input to the D8 model is the state of the UAV containing the elevation and the water level information of the current and neighbouring cells (8 adjacent neighbours).

The output of the D8 algorithm is a flow direction corresponding to each cell that helps in realizing a better exploration policy for our proposed model. The cell with the highest water accumulation as given by Equation 2 is calculated by analyzing the discharge of water at each cell, derived by applying Manning's equation [16]. The hydrological model for flowing water is based on the Saint Venant conditions [12] (for more details see [11, 12, 16]).

$$L_t^{\eta_{i_c}} = \underset{c_n^{\eta_i} \in n}{argmax} \left(\frac{\left(M_{t+\Delta t}^{\eta_{i_c}} - M_{t+\Delta t}^{c_n^{\eta_i}} \right) \cdot frate}{d(\eta_{i_c}, c_n^{\eta_i})} \right) \tag{2}$$

$$M_{t+\Delta t}^{\eta_{i_c}} = \frac{\left(w_t^{\eta_{i_c}} + ({}^{in}M_t^{\eta_{i_c}} - {}^{out}M_t^{\eta_{i_c}}) \frac{\Delta t}{\Delta \eta_{i_c}} \right)^{\frac{5}{3}}}{\varphi} \tag{3}$$

where, η_{i_c} is the cell occupied by a UAV η_i at time t. $c_n^{\eta_i}$ denotes the neighbouring cells to η_{i_c} from the set n of 8 possible neighbours. $L_t^{\eta_{i_c}}$ denotes the cell with the lowest relative water discharge in the neighbourhood of η_{i_c}. $M_{t+\Delta t}^{\eta_{i_c}}$ calculates the water discharge for the cell c that is currently under the observation of UAV η_i. φ denotes the Manning roughness coefficient. $w_t^{\eta_{i_c}}$ depicts the water depth at cell c at time t.

The cell with the lowest water discharge is usually the one with the highest water accumulation which puts it relatively at a higher risk than others. Once we have identified a neighbouring cell with the lowest water discharge using D8, this information is used to generate an exploration action for the UAV to move in the estimated flow direction. There lies an exception where the UAV η_i may decide to hover over the same cell when cell η_{i_c} itself is the one with the highest water accumulation.

Equation 4 denotes the exploration action given by D8 flow estimation technique:

$$a_{D8_t}^{\eta_{i_c}} = W(L_t^{\eta_{i_c}}, \eta_{i_c}) \tag{4}$$

where $a_{D8_t}^{\eta_{i_c}}$ represents the action of a UAV η_i based on the D8 flow algorithm. The function $W(.,.)$ maps the appropriate action from the feasible action set A^{η_i} (see Section 4).

4 PROPOSED METHOD:DEC-DQNC8

In this section, we define the dec-POMDP problem that is considered in this paper. Having a system of n UAVs: $N; \{\eta_i | i = 1, 2, ..., n\}$, the objective is to capture as many critical regions (of higher risk levels) as possible by performing decentralized area coverage under the constraint of limited batteries of the UAVs with minimum overlapping of UAV trajectories.

The flood environment is partially observable as each UAV is only able to perceive its local surroundings and the information communicated by other UAVs (via opportunistic communication) and is unaware of the global state. For our multi-UAV system, the dec-POMDP is given by a tuple $< N, S, \{A^{\eta_i}\}, P^T, \{O^{\eta_i}\}, \{R^{\eta_i}\}, P^O, \mathbb{H} >$ where N is the set of UAVs and S denotes a finite set of hidden states. $A^{\eta_i} = \{N, S, E, W, NE, NW, SE, SW, hover, hover + comm\}$ (comm is the abbreviation for communication) is the finite action set. P^T is the state transition probability that provide the distribution $P^T(s_{t+1}|s_t, a_t)$ of transitioning to the next state s_{t+1} given the current state s_t and joint action $a_t = \{a^{\eta_1}, a^{\eta_2}, ..., a^{\eta_n}\} \in \mathbb{A}$. $\mathbb{A}$ is the joint action space of all the UAVs. $O^{\eta_i} = \{o_1^{\eta_i}, o_2^{\eta_i}, ...\}$ is the finite observation set for each UAV η_i, where a single observation is represented as:

$$o_t^{\eta_i} : \{p_{l_{x+k_1, y+k_2}}^{c^{\eta_i}}, e_{x+k_1, y+k_2}^{c^{\eta_i}}, \mathbb{L}_c | \ \forall \ k_1, k_2 \in \{-1, 0, 1\}\}$$

where $p_{l_{x,y}}^c$ corresponds to the human population density level of cell c and $e_{x,y}^c$ corresponds to the terrain elevation. $\mathbb{L}_c$ is the water level at the sensed location c. The $k1, k2 \in \{-1, 0, 1\}$ set represents the elevations and human population density levels of the neighbouring 8 cells of c. P^O is the observation probability given the conditional probability $P^O(o_{t+1}|s_{t+1}, a_t)$. Set R^{η_i} defines the accumulated rewards earned by each UAV $\forall \ i \in N$ present in the system. We use $\mathbb{H}$ to indicate a decision horizon.

At each time-step t, the UAV takes an action a^{η_i} based on its local observation history $o_{[1:t]}^{\eta_i}$ and receives the next observation $o_{t+1}^{\eta_i}$ and rewards $R_t^{\eta_i}$ from the environment. As the complete state of the environment is unknown to the UAV, the observation history is useful in realizing a local policy π^{η_i}. So, π^{η_i} can be defined as the mapping from local histories to agent-specific actions and the joint policy $\Pi : \{\pi^{\eta_i}, \pi^{\eta_i}, ..., \pi^{\eta_N}\}$ is the set of local policies corresponding to each agent $\eta_i \in N$.

Collision Avoidance: Further, to implement the protocol for collision avoidance among the UAVs, an overlapping constraint (OC) is defined to discourage the UAVs from getting into the collision-prone range of each other. The overlapping constraint is given as:

$$OC_t(\eta_i, \eta_j) = \begin{cases} 1 & if \ \ d_t^{\eta_i, \eta_j} \leq T_c \ \forall \ \eta_i, \eta_j \in N \\ 0 & otherwise \end{cases} \tag{5}$$

where, $d_t^{\eta_i, \eta_j}$ is the observed Euclidean distance between the two UAVs (in square meters). T_c is the threshold distance below which the UAVs are prone to collision. A penalty is imposed on the UAV when its action leads it into a collision prone range with other UAV(s). The collision prone areas are calculated in reference to the coverage maps (CM's) communicated among the UAVs (discussed in section 4.4). As we are predicting future locations of the UAVs based on their CM's, a UAV analyzes a static path foreseeing a

possible collision course with other UAVs (the ones with which it has communicated).

UAV Energy Model: A UAV requires energy for various manoeuvres such as take-off, hovering and flying in addition to the energy required for transmission. Authors in [32] derived an analytical model for propulsion power consumption $\mathbb{W}$ of quadrotor UAVs moving/flying at a speed of $\mathcal{V}$. In [37], authors discuss the energy required to realize the communication transmission between UAVs. Adopting the energy consumption model from [32] and [37], a UAV's energy λ_t at any given time t can be derived as:

$$\lambda_t = \lambda_{t-1} - \lambda_{\{hover,moving,comm\}} \tag{6}$$

$$\lambda_{moving} = \mathbb{W}(\mathcal{V}) \times \mathbb{T}_{moving} \tag{7}$$

where $\mathbb{T}_{moving}$ is the number of time-steps during which the UAV is in motion.

$$\lambda_{hover} = \mathbb{W}(0) \times \mathbb{T}_{hover} \tag{8}$$

where $\mathbb{T}_{hover}$ is the number of time-steps during which the UAV hovers.

$$\lambda_{comm} = \Omega \times \mathbb{T}_{comm} \tag{9}$$

where Ω is the transmission power of the UAV and $\mathbb{T}_{comm}$ is the number of time-steps during which the UAV is transmitting data.

Energy is not part of the reward formulation as the objective is focused on coverage only, however, it plays a crucial role by affecting the episode length/UAV flight time. It limits the UAV actions based on the current energy level of the UAV. For example, if $\lambda_t^{\eta_i} < \lambda_{comm}$ the UAV η_i cannot perform communication anymore, similarly for other actions $\{hover, moving\}$.

4.1 The Value and Reward Functions

The objective of the multi-UAV system is to find the optimal joint policy (Π) that achieves maximum cumulative rewards in the long run. The state-action value function $Q^{\Pi}(s_t, a_t)$ under the policy Π defines the long-term desirability of an action in a particular state, given as

$$Q^{\Pi}(s_t, a_t) = \mathbb{E}_{a_t \sim \Pi}\left[\sum_{k=0}^{\mathbb{H}} \gamma^k R_{t+k+1} | s_t, a_t\right] \tag{10}$$

where $0 \geq \gamma < 1$ is the discount factor (used to maintain finite sum over the infinite horizon).

In the considered scenario, as the agents are unaware of the global state, the Q-value is defined in terms of the global expected utility as the sum of the local utilities of each UAV.

$$Q(O_t^N, a_t) = \sum_{\eta_i, \eta_j \in N} Q^{\eta_i}(o_{[1:t]}^{\eta_i}, a_t^{\eta_i}, a_t^{\eta_j}) \tag{11}$$

where O_t^N is the joint observation set at time t, a_t is the joint action. $o_{[1:t]}^{\eta_i}$ is the local observation history of UAV η_i up until time t. $a_t^{\eta_i}$ corresponds to the local action of i^{th} UAV at time t and $a_t^{\eta_j}$ denotes the action of agents ($\eta_j \in N$) that interacted with the UAV η_i at time t (a UAV can communicate with only a single UAV at any given time).

The formulation of the proposed reward function is based on the information gained by the UAV from the environment. The other parameter is whether the UAV communicating or not and whether it's on a collision course with another UAV or not. The information gain is a quantitative measure that defines the importance of the cell c captured by the UAV. Higher the risk level of a cell, higher is the information gain. Hence the information gain ($I_c^{\eta_i}$) from a cell c as observed by a UAV η_i is given as:

$$I_c^{\eta_i} = \frac{\mathcal{U}_l^c}{\mathcal{U}_l^{max}} \tag{12}$$

where $\mathcal{U}_l^c$ is the risk level of cell c and $\mathcal{U}_l^{max}$ is the maximum possible risk level. The reward $R_t^{\eta_i}$ corresponding to an individual UAV η_i at time t is calculated as:

$$R_t^{\eta_i}(s_t^{\eta_i}, a_t^{\eta_i}, s_{t+1}^{\eta_i}) = I_c^{\eta_i} + C_c^{\eta_i} - \alpha_t(\eta_i, \eta_j).OC_t(\eta_i, \eta_j) \quad \forall\ \eta_j \in N, j \neq i \tag{13}$$

where $C_c^{\eta_i}$ is a positive scalar incentive given to η_i provided that it is successfully able to build a communication link with another UAV at time t. $\alpha_t(.,.)$ is a function that outputs a scalar penalty when a UAV η_i's action leads it into a collision-prone range. Penalty corresponding to function $\alpha(.,.)$ is greater than $C_c^{\eta_i}$ as we do not prioritize UAV communication at the cost of a collision. Function (OC) denotes whether two UAVs are within the collision range or not (the communication range is greater than the collision range).

4.2 Action Selection

In the standard DQN model [24], the action selection is based on the ϵ-greedy strategy. As initially, the environment is completely unknown, random actions are performed to explore the value of different actions from various states. In later stages of training, the agent exploits the already gathered information to perform the given task in an optimal manner. However, random exploration can lead to a sub-optimal policy as it is the least efficient exploration method when it comes to limited energy models [23]. In our proposed approach nicknamed dec-DQNC8, we opt for a better exploration strategy based on the D8 flow estimation algorithm. The employed action selection strategy (for more details see [11]) is given as:

$$a_t^{\eta_i} = \begin{cases} \underset{a'}{argmax}\ Q(o_{[1:t]}^{\eta_i}, a') & 1 - (\epsilon_1 + \epsilon_2)\ probability \\ a_{D8_t}^{\eta_{i_c}} & \epsilon_2\ probability \\ random\ \ action & \epsilon_1\ probability \end{cases} \tag{14}$$

where, $0 \leq \epsilon_1, \epsilon_2 \leq 0.5$

η_i denotes the i^{th} UAV and a' denotes the action given by the target network of DQN. $a_{D8_t}^{\eta_{i_c}}$ denotes the action based on D8 flow estimation algorithm. $o_{[1:t]}^{\eta_i}$ is the local observation history of UAV η_i up until time t. UAV's action ($a_t^{\eta_i}$) is subjected to its available energy (refer section 4). The proposed dec-DQNC8 is depicted in Algorithm 1. As can be noted from line number 1-3, We randomly initialized the system related hyperparameters and the network

weights. Next, within each episode, the observations of each UAV w.r.t. the captured image of the environment is recorded and marked in individual coverage maps (refer to line number 4-6). Then an action is performed by the UAV(s) based on its current action policy, as depicted in line number 8. Later, when enough experience is gathered by local movements and experience gained from communication, the policy network is updated by minimizing the loss and shifting the gradient in the correct direction (refer to line number 11-18).

Algorithm 1: Proposed dec-DQNC8 Algorithm

1 **Input:** n, Ω, f_c, $\mathcal{V}$, γ [1]
2 Initialize action value function Q^{η_i} with random weights θ^{η_i} for each agent $i \in N$
3 Initialize target action value function $\hat{Q}^{\eta_i}$ with weights $\theta^{-} = \theta^{\eta_i}$ for each agent $i \in N$
4 **for** *UAV* = $\eta_i, \eta_j, ..., \eta_N$ **do**
5 **for** *episode=1,2,...* **do**
6 Load initial observation o^{η_i} and coverage map $\hat{m^{\eta_i}}$
7 **while** $t \leq$ *max_time_step and* $\lambda_t > \lambda_{\{hover,moving,comm\}}$ **do**
8 Select action $a_t^{\eta_i}$ as given in Equation 14, Section 4.2
9 Decay UAV's energy as given in Equation 6, Section 4
10 UAV η_i makes the next observation $o_{t+1}^{\eta_i}$ and receives reward $R_t^{\eta_i}$ from the environment.
11 Store transition $< o_t^{\eta_i}, a_t^{\eta_i}, R_t^{\eta_i}, o_{t+1}^{\eta_i} >$ in replay buffer $\mathcal{Z}^{\eta_i}$
12 **if** $a_t^{\eta_i} ==$ *hover + comm* **then**
13 $\mathcal{Z}^{\eta_i}.append(\mathcal{Z}^{\eta_j})$ // Considering η_j and η_i communicated at time t
14 Update the input state as given in Equation 19, Section 4.4
15 **end**
16 Sample a random mini-batch of $\mathcal{B}$ transitions $(o_k^{\eta_{i'}}, a_k^{\eta_{i'}}, R_k^{\eta_{i'}}, o_{k+1}^{\eta_{i'}})$ from $\mathcal{Z}^{\eta_i}$, where k denotes the index of $\mathcal{B}$ and $\eta_{i'}$ denotes the experience of η_i or the experience of any other UAV that communicated with η_i.
17 Calculate target Q value: $y_t^{\eta_i} = R_k^{\eta_i'} + \gamma \max_{a'} \hat{Q}(o_k^{\eta_i'}, a'; \theta^{-})$
18 Perform a gradient decent step on $(y_t^{\eta_i} - Q(o_k^{\eta_i'}, a_k^{\eta_i'}; \theta))$ w.r.t. network parameter θ in every C steps and reset $\hat{Q} = Q$, i.e., set $\theta^{-} = \theta$
19 **end**
20 **end**
21 **end**

4.3 UAV-to-UAV Communication

A UAV η_i can transmit the replay buffer $\mathcal{Z}^{\eta_i}$ and coverage map $\hat{m}^{\eta_i}$ in form of packets to another UAV η_j if the UAVs are within a transmission range d_τ of one another [14]. It is assumed that the UAVs are equipped with radio transmitters and work on 2.4 GHz to 5.8 GHz frequency. This range is calculated as:

$$d_\tau = 10^{-[10\times log_{10}(\zeta^{\eta_i\eta_j}\Upsilon/\Omega)+28+20\times log_{10}(f_c)]/22} \tag{15}$$

where $\zeta^{\eta_i\eta_j}$ is the threshold SNR. Ω is the transmission power of the UAV and Υ denotes the thermal noise described as the Gaussian white noise [2]. $h^{\eta_i\eta_j}$ is the fading coefficient of communication link between UAV η_i and η_j based on Nakagami-m distributions [27]. f_c is the carrier frequency of the communication signal [21]. A packet is successfully received if the mean SNR $\hat{\zeta}^{\eta_i\eta_j}$ is greater than the threshold SNR $\zeta^{\eta_i\eta_j}$.

$$\hat{\zeta}^{\eta_i\eta_j} = \frac{\Omega}{\Upsilon} \times 10^{-\frac{28+22\times log_{10}(d^{\eta_i\eta_j})+20\times log_{10}(f_c)}{10}} \tag{16}$$

Whenever two UAVs (η_i, η_j) are within a distance d_τ of each other, both the UAVs perform *hover + comm* action to transmit the replay buffer information and their coverage maps to each other (conditioned on whether enough energy is left for transmission or not). After transmission completes, UAV η_i's replay buffer becomes equal to $\mathcal{Z}^{\eta_i}.append(\mathcal{Z}^{\eta_j})$ (and similarly for UAV η_j). This increase in information helps the DQN to converge early and the resultant policy is more robust as it is learnt from a more diverse and distributed form of data. As the replay buffer data is shared among UAVs over communication, the individual policies learnt by the UAVs could be overlapping to some extent. Still, it would be very rare for two (or more) UAVs to end up with very similar weights at the end of training. Since the initial weights of DQN models are different (as randomly initialized) and also it would be very uncommon of two (or more) UAVs to have exactly the same data to train from during each episode.

4.4 Coverage Maps

In this section, we propose the use of a coverage map (*CM*) that contains the local trace of the UAV based on its local observation history. This map is also shared among the UAVs when they communicate. A coverage map contains the information corresponding to the locations that a UAV has visited during its flight and also the recent time-step at which that particular cell was observed. The data contained in the map is used to update the information gain effectively altering the rewards that a UAV can accumulate from a cell (this is applicable to the time steps after the communication has occurred). The updated information gain is calculated as:

$$I_c^{\eta_i} = \frac{\mathcal{U}_l^c}{\mathcal{U}_l^{max}} \cdot \frac{t_c^{\eta_i} - t_c^{\eta_j}}{t_c^{\eta_i}} \tag{17}$$

In reference to the UAV η_i currently observing cell c, the updated information gain $I_c^{\eta_i}$ is calculated corresponding to $t_c^{\eta_i}$ (the current time-step) and $t_c^{\eta_j}$ ($\eta_j \in \mathbb{N}$, the UAVs that have communicated and transferred their coverage maps to η_i up until time t). $t_c^{\eta_j}$ denotes the last time-step at which the cell c was observed, as recorded in the *CM*. The idea is to diminish the rewards of a UAV for the cells that have been previously visited by the other UAVs. This helps

to learn a local policy by a UAVs to emphasizes on visiting more unobserved cells rather than revisiting the observed ones.

We further estimate the future locations of a UAV (in reference to η_j) based to its map $\hat{m}_t^{\eta_j}$. It is calculated as follows:

$$C_{t+1}^{\eta_j} = \underset{c_n^{\eta_j}}{argmax}(M_t^{\eta_j} \mid \hat{m}_t^{\eta_j}) \tag{18}$$

where, $C_{t+1}^{\eta_j}$ represents the location of UAV η_j at time-step $t+1$. $\underset{c_n^{\eta_j}}{argmax}(.|.)$ depicts the cell with the highest water accumulation (usually the lowest elevation cell) in the neighbourhood ($c_n^{\eta_j}$) of η_j. This information can also be extracted from the shared experience replay buffer, but it's not necessary that this information will be selected in every mini-batch sampled for training. So to make sure that the communicated trace information is available to the UAVs during training (at all times), we explicitly include the coverage map information as state input to the model. A coverage map also has a huge impact when the model's performance is evaluated in a test environment (discussed in Section 5.3 and 5.4).

The updated information gain as given in Equation 17 only corrects the future rewards considering the cells that are being revisited. However, we also need to correct the already observed rewards that are in the replay buffer. Let's say at time t UAV η_i communicated with η_j (a UAV can communicate with only a single UAV at a time), the replay buffer of η_i becomes equal to $\mathcal{Z}^{\eta_i}.append(\mathcal{Z}^{\eta_j})$. Based on the number of overlapping cells between η_i and η_j up until time t as observed from the traces in $\hat{m}^{\eta_i}$ and $\hat{m}^{\eta_j}$, the rewards in the replay buffer are updated in reference to the Equations 13 and 17. For a UAV η_i the updated state contains the local state information, its coverage map and the coverage map of the UAVs with which η_i communicated.

$$s_{input}^{\eta_i} = \{o^{\eta_i}, \{\hat{m}^{\eta_i}, \hat{m}^{\eta_j}\}\}\ j \in N\ ;\ j \neq i \tag{19}$$

5 EXPERIMENTATION AND RESULTS

In this section, we discuss and analyze the performance of our proposed model dec-DQNC8 with the simpler variant dec-DQN8 and two other state-of-the-art multi-UAV coverage techniques from the literature, namely, dec-DQN [7] and PSO [31]. The replay buffer updates to rewards on communication are not applied in the case of dec-DQN8 and the use of coverage maps is also not-opted for dec-DQN8. In [7], authors employ a DQN-based approach to learn multi-UAV controls for flood monitoring. The proposed technique is said to be decentralized but no communication method is introduced or employed. During comparison with the proposed model, we treat the algorithm given by [7] as decentralized DQN with no communication, where the UAVs are only aware of their local state. In [31], the authors proposed a distributed PSO model to perform exploration of a disaster area using UAVs. There is no central node considered in this distributed approach, but the UAVs are able to share their local information with other UAVs that are in their vicinity.

To implement our proposed model, we consider a team of 7 UAVs (quadrotors) deployed during various experiments. The altitude of the UAVs is fixed at 100 meters above sea level. Considering a standard IRIS UAV, the UAV camera angles are assumed to be 45 degrees and 30 degrees. To create the training environment, the coastal region of Chennai city (a city in the state of Tamil Nadu, India) is selected and its elevation data is accessed using the Topographic map tool [29]. We also consider a different test environment to evaluate the learnt policies of the multi-UAV system. Further, the information on the water level of each location is encoded using the FloodMap application [3]. The water level (w_l) information is defined at 8 levels, ranging from $\{1ft - 8ft\}$. Human population density information is gathered using the Flood Mapping tool [1]. The collision range for UAVs is set to 10 meters, under which UAVs are bound to collide. Implementation is done on Google Colab having, Intel(R) Xeon(R) CPU, 2.30GHz CPU frequency, and 12GB RAM. We make use of the following performance metrics to test our proposed model:

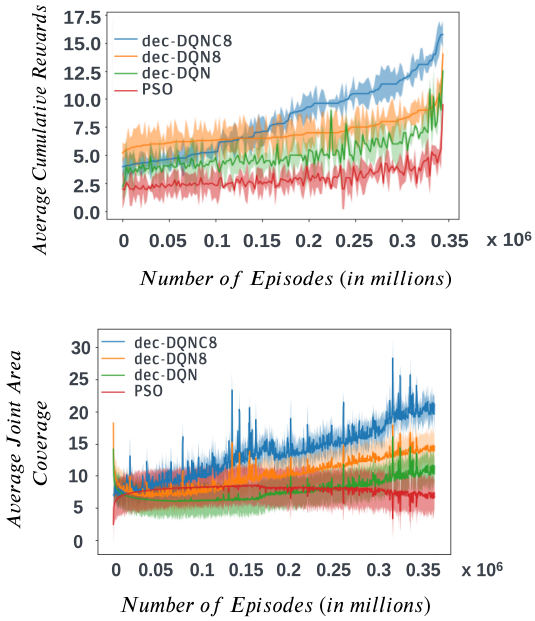

Figure 1: Performance comparison between dec-DQNC8, dec-DQN8, dec-DQN and PSO based on (a) average cumulative rewards and (b) average joint coverage during training.

- Average cumulative rewards observed by the multi-UAV system over the training and test environments.
- Average joint coverage observed during training and testing.
- Multi-UAV path trace observed over the training and test environments.

Results are observed over 5 different random seeds and the deviation around the mean is highlighted in the plots.

5.1 Average Cumulative Rewards Observed During Training

Experiments are performed over the course of 0.35 million episodes to observe the difference in cumulative rewards more vividly (if present). Each episode is of 1000 time steps. Figure 1(a) depicts the average cumulative rewards observed by various models during

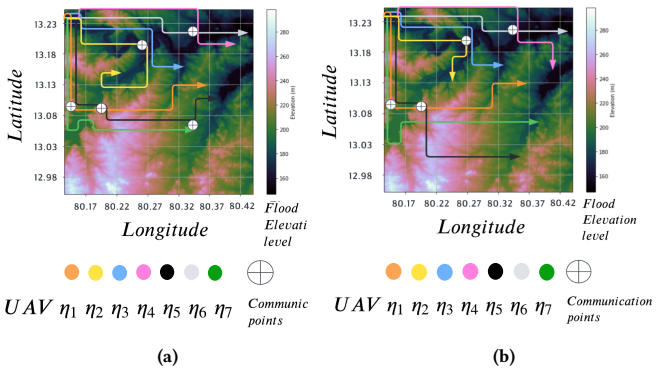

Figure 2: The multi-UAV path trace observed during training with (a) No updation in rewards and (b) Updating rewards within episodes in the experience replay buffer.

training. As observed, all the RL-based methods perform similarly in the initial episodes with dec-DQN8 performing the best. But, as the number of episodes progresses dec-DQNC8 outperforms the others to achieve the highest overall rewards. Such behaviour of dec-DQNC8 is justified as the rewards are updated within episodes whenever the UAV communicates, leading to a decline in average rewards. However, as soon as the coverage map becomes good enough after communication, UAVs are able to spread over the environment in a much better way, covering a significantly larger area. PSO performs the worst highlighting the pitfall of PSO as it usually gets stuck in local optima, especially when training is done using a limited number of environment parameters.

5.2 Average Joint Coverage Observed During Training Along with the Multi-UAV Path Trace

In this experiment, we observed the average coverage of the environment by the multi-UAV system during training. The coverage is defined as the number of unique cells captured by the UAVs in an episode. Higher coverage provides a better chance of capturing a relatively larger number of critical regions. Figure 1(b) depicts the average coverage by the multi-UAV system during training. As observed, up until the 0.15 million episodes there is no significant change in coverage to separate the models, but after 0.2 million (approx.) episodes dec-DQN shows a noticeable improvement in the joint coverage as it sees a linear rise, outperforming PSO. A noticeable gap in joint coverage can be observed between dec-DQNC8 and dec-DQN8 after 0.25 million episodes and this gap seems to increase with the increase in the number of episodes. This signifies the impact of the coverage map used by dec-DQNC8 in achieving a better spread over the environment.

To further analyse the impact of updating rewards and sharing coverage maps in learning a multi-UAV policy, we trace the paths of the UAVs during training for both dec-DQNC8 and dec-DQN8. dec-DQN8 adopts the D8 flow estimation strategy for better exploration and shares the experience replay buffer across UAVs during communication. But, it does not update the rewards of the replay buffer on

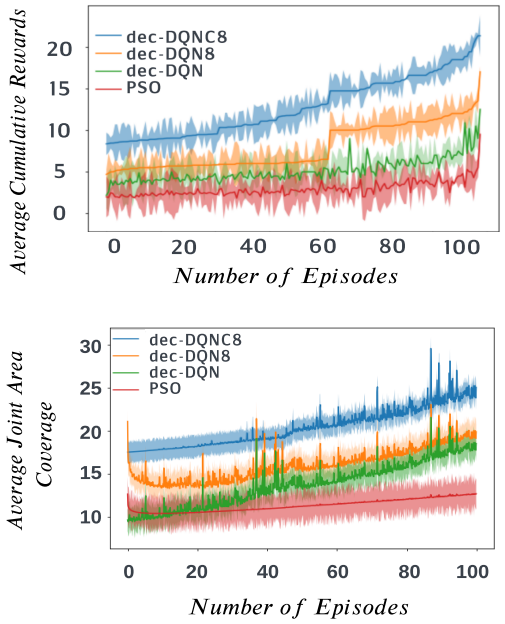

Figure 3: Performance comparison between dec-DQNC8, dec-DQN8, dec-DQN and PSO based on (a) average cumulative rewards and (b) average joint coverage during testing.

communication and also does not use the coverage map for policy learning. Figure 2 illustrates the multi-UAV path observed during a single episode at the end of the training by both dec-DQNC8 and dec-DQN8. As observed, the updation in rewards and the use of *CM* helps in maintaining better coverage over the environment in the long run. Rather than forcing a constraint on the multi-UAV system to maintain inter-UAV separation, here the system learns naturally about the effects of clustering with other UAVs.

5.3 Average Cumulative Rewards Observed During Testing

To analyze the generalizability and robustness of the learnt policy using dec-DQNC8, the performance of the model is observed in a test environment over 100 episodes. During testing, the learnt policies are not updated, but the UAVs are able to share their coverage maps with each other during communication (in the case of dec-DQNC8). The test environment is simulated using the real-world elevation data of the Barpeta district of Assam which is one of the most prone regions to floods. Figure 3(a) depicts the average cumulative rewards observed by various algorithms during testing. As observed, dec-DQNC8 has the best performance from the initial episode itself. This highlights the fact that the proposed model is able to learn a robust multi-UAV policy. dec-DQN8 also achieves significant rewards during testing. This justifies that the D8-based models are able to learn a better policy and achieve relatively higher rewards even in a short duration of time as compared to dec-DQN. PSO and dec-DQN have similar performances, with PSO performing the worst in later episodes.

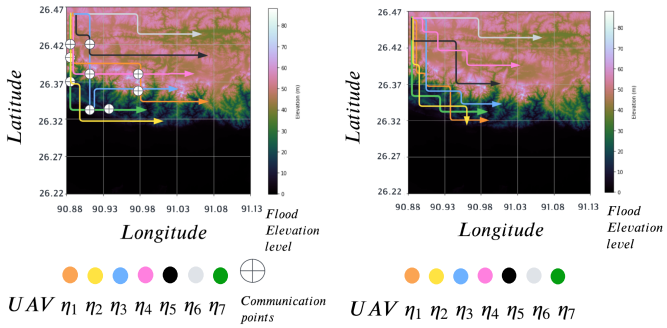

Figure 4: The multi-UAV path trace observed during testing with (a) allowable inter-UAV communication and (b) no communication.

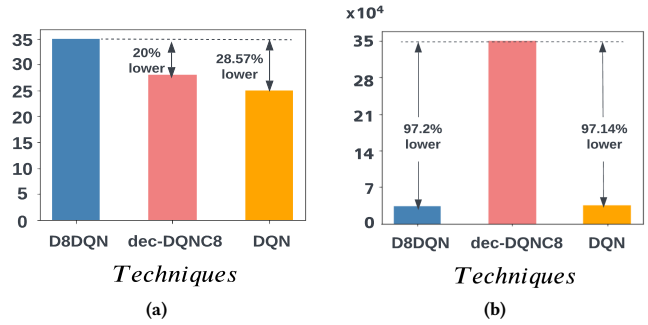

Figure 5: Centralized (D8DQN) vs Decentralized (dec-DQNC8) system comparison based on (b) average joint coverage (b) Number of episodes to converge during training.

5.4 Average Joint Coverage Observed During Testing Along with the Multi-UAV Path Trace

In this experiment, we analyze the average coverage of the test environment by the multi-UAV system. This helps in highlighting the difference in the learnt policies and to observe whether the multi-UAV system is able to maintain a considerable spread over the environment or not when the policies are fixed. Figure 3(b) depicts the average joint coverage observed by the UAVs in the test environment. As observed, the proposed model dec-DQNC8 sees better coverage as compared to other algorithms. This highlights the impact of using the coverage map to learn the policies by including the map as part of the input state. dec-DQN8 sees better coverage as compared to dec-DQN up until 60^{th} episode. PSO sees the worst performance, highlighting the difficulty of learning in a highly dynamic environment.

To analyze the significance of communication during testing, we considered two scenarios, one where the UAVs can communicate with each other and the other where the communication was restricted. Figure 4 represent these two scenarios where the multi-UAV path trace was observed during testing. As can be seen, if the UAVs are allowed to communicate they are able to spread significantly better over the test environment just by sharing their coverage maps. This results in a higher chance of covering a larger number of critical regions.

5.5 Comparison with Centrally trained policies

For the performance gap between centralized and decentralised policies, our proposed approach dec-DQNC8 is compared with D8DQN and DQN [11] over the average joint area coverage and the number of episodes required to converge during training, as given in Figure 5(a) and 5(b) respectively. All three models are trained and observed under identical environmental conditions. As depicted in Figure 5(a), the centralized approach (D8DQN) achieves 20 % larger coverage as compared to dec-DQNC8. Such an outcome can be intuitively analyzed since in the case of D8DQN all the UAVs are jointly regulated by the central system whereas, in the case of dec-DQNC8, the UAVs are dependent on experience sharing via opportunistic communication. However, the effect of utilizing domain knowledge using D8 has a significant impact on the models, since dec-DQNC8 achieves 8.5 % larger area coverage than the standard DQN model (i.e., a centralized approach). In terms of convergence, all three approaches do converge but at different rates. D8DQN and DQN only took about 2.8 % (approx.) of the number of episodes to converge as compared to dec-DQNC8. Such a large gap signifies the difficulty in training a decentralized model as compared to a centralized one. But knowing that a decentralized based approach also converges is a step towards realizing real-world models.

6 CONCLUSION

This paper presents a decentralized deep reinforcement learning algorithm, known as dec-DQNC8 that is used to learn multi-UAV controls to perform coverage of flooded regions. UAVs are tasked to perform time-sensitive area coverage where the UAV's energy depletes with every time step and varies based on the action the UAV is performing. The objective is to capture as many critical regions as possible under the limited energy constraint. In this sense, the task of area coverage needs to be addressed globally by multiple UAVs to maximize the coverage of the environment. The proposed model is fully decentralized as no central entity is considered to achieve the task of area coverage using multiple UAVs cooperatively. UAVs are able to communicate with other UAVs given that they are present within the transmission range. dec-DQNC8 utilizes domain knowledge for policy learning by employing the D8 flow estimation algorithm that improves the exploration strategy of our DRL based method. Further, the proposed model also uses a coverage map that contains the path trace of individual UAV's. We also learn the coverage probability of neighbouring cells based on the information contained in the map to cover a significantly larger area and avoid revisiting observed cells. Results are obtained over the training and test environments and the observations signify the impact of the proposed algorithm (dec-DQNC8) in realizing a successful decentralized multi-UAV policy, as noticeable improvements are observed across different performance metrics.

ACKNOWLEDGMENTS

The first author would like to thank TCS for their support under the TCS Research Scholar Program.

REFERENCES

[1] 1985. *Flood Mapping Tool.* https://floodmapping.inweh.unu.edu Accessed: 2022-06-13.

[2] 2015. *What is snr?* https://mason.gmu.edu/~rmorika2/What_is_SNR_.htm Accessed: 2022-06-13.

[3] 2020. *Flood map: Water level elevation map.* https://www.floodmap.net/ Accessed: 2022-06-13.

[4] 2022. *Chennai topographic map, elevation, relief.* https://en-in.topographic-map.com/maps/fbim/Chennai/ Accessed: 2022-06-13.

[5] Mohamed Abdelkader, Samet Güler, Hassan Jaleel, and Jeff S. Shamma. 2021. Aerial swarms: Recent applications and challenges. *Current Robotics Reports* 2, 3 (2021), 309–320. https://doi.org/10.1007/s43154-021-00063-4

[6] Ahmad Azar, Anis Koubaa, Nada Mohamed, Habiba Ibrahim, Zahra Fathy, Muhammad Kazim, Adel Ammar, Bilel Benjdira, Alaa Khamis, Ibrahim Hameed, and Gabriella Casalino. 2021. Drone Deep Reinforcement Learning: A Review. *Electronics* 10 (04 2021). https://doi.org/10.3390/electronics10090999

[7] David Baldazo, Juan Parras, and Santiago Zazo. 2019. Decentralized Multi-Agent Deep Reinforcement Learning in Swarms of Drones for Flood Monitoring. In *2019 27th European Signal Processing Conference (EUSIPCO)*. 1–5. https://doi.org/10.23919/EUSIPCO.2019.8903067

[8] Guillaume Bono, Jilles Steeve Dibangoye, Laëtitia Matignon, Florian Pereyron, and Olivier Simonin. 2019. Cooperative multi-agent policy gradient. *Machine Learning and Knowledge Discovery in Databases* (2019), 459–476. https://doi.org/10.1007/978-3-030-10925-7_28

[9] Lucian Busoniu, Robert Babuska, and Bart De Schutter. 2008. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38, 2 (2008), 156–172.

[10] M. L. Cummings. 2014. Operator interaction with centralized versus decentralized UAV architectures. *Handbook of Unmanned Aerial Vehicles* (2014), 977–992. https://doi.org/10.1007/978-90-481-9707-1_117

[11] Armaan Garg and Shashi Shekhar Jha. 2022. Directed Explorations During Flood Disasters Using Multi-UAV System. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*.

[12] William S. Gonwa and M. Levent Kavvas. 1986. A modified diffusion equation for flood propagation in trapezoidal channels. *Journal of Hydrology* 83, 1 (Jan. 1986), 119–136. https://doi.org/10.1016/0022-1694(86)90187-3

[13] Jayesh K. Gupta, Maxim Egorov, and Mykel Kochenderfer. 2017. Cooperative multi-agent control using Deep Reinforcement Learning. *Autonomous Agents and Multiagent Systems* (2017), 66–83. https://doi.org/10.1007/978-3-319-71682-4_5

[14] Hamed Hellaoui, Ali Chelli, Miloud Bagaa, and Tarik Taleb. 2020. UAV Communication Strategies in the Next Generation of Mobile Networks. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*. 1642–1647. https://doi.org/10.1109/IWCMC48107.2020.9148312

[15] Pin-Chun Huang. 2020. Analysis of Hydrograph Shape Affected by Flow-Direction Assumptions in Rainfall-Runoff Models. *Water* 12, 2 (2020). https://doi.org/10.3390/w12020452

[16] Pin-Chun Huang and Kwan Tun Lee. 2013. An efficient method for DEM-based overland flow routing. *Journal of Hydrology* 489 (2013), 238–245. https://doi.org/10.1016/j.jhydrol.2013.03.014

[17] Prakarsh Kaushik, Armaan Garg, and Shashi Shekhar Jha. 2021. On Learning Multi-UAV Policy for Multi-Object Tracking and Formation Control. In *2021 IEEE 18th India Council International Conference (INDICON)*. 1–6. https://doi.org/10.1109/INDICON52576.2021.9691567

[18] Mikko Lauri, Joni Pajarinen, and Jan Peters. 2019. Information Gathering in Decentralized POMDPs by Policy Graph Improvement. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '19)*. International Foundation for Autonomous Agents and Multiagent Systems, 1143–1151.

[19] Yongheng Liang, Hejun Wu, and Haitao Wang. 2022. ASM-PPO: Asynchronous and Scalable Multi-Agent PPO for Cooperative Charging. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS '22)*. International Foundation for Autonomous Agents and Multiagent Systems, 798–806.

[20] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6382–6393.

[21] Hendrik Lumbantoruan and Koichi Adachi. 2021. Array antenna equipped UAV-BS for efficient low power WSN and its theoretical analysis. *IET Communications* 15, 16 (2021), 2054–2067. https://doi.org/10.1049/cmu2.12238

[22] Xueguang Lyu, Yuchen Xiao, Brett Daley, and Christopher Amato. 2021. Contrasting Centralized and Decentralized Critics in Multi-Agent Reinforcement Learning. In *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*. ACM, 844–852.

[23] Roger McFarlane. 2003. A Survey of Exploration Strategies in Reinforcement Learning. https://www.cs.mcgill.ca/~cs526/roger.pdf

[24] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. *arXiv e-prints*, Article arXiv:1312.5602 (Dec. 2013), arXiv:1312.5602 pages. https://doi.org/10.48550/ARXIV.1312.5602 arXiv:1312.5602

[25] Hafiz Suliman Munawar, Ahmed W.A. Hammad, and S. Travis Waller. 2022. Disaster Region Coverage Using Drones: Maximum Area Coverage and Minimum Resource Utilisation. *Drones* 6, 4 (2022). https://doi.org/10.3390/drones6040096

[26] Frans A Oliehoek. 2012. Decentralized pomdps. In *Reinforcement Learning*. Springer, 471–503.

[27] Tharindu D. Ponnimbaduge Perera, Dushantha Nalin K. Jayakody, Sahil Garg, Neeraj Kumar, and Ling Cheng. 2020. Wireless- Powered UAV assisted Communication System in Nakagami-m Fading Channels. In *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*. 1–6. https://doi.org/10.1109/CCNC46108.2020.9045123

[28] Alejandro Puente-Castro, Daniel Rivero, Alejandro Pazos, and Enrique Fernandez-Blanco. 2022. UAV swarm path planning with reinforcement learning for field prospecting. *Applied Intelligence* (2022). https://doi.org/10.1007/s10489-022-03254-4

[29] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani. 2019. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* 7 (2019), 48572–48634. https://doi.org/10.1109/ACCESS.2019.2909530

[30] Richard S. Sutton and Andrew Barto. 2018. *Reinforcement learning: an introduction*. The MIT Press.

[31] Jesús Sánchez-García, Daniel Gutiérrez, and S.L. Toral. 2018. A distributed PSO-based exploration algorithm for a UAV network assisting a disaster scenario. *Future Generation Computer Systems* 90 (07 2018). https://doi.org/10.1016/j.future.2018.07.048

[32] Yong Zeng, Jie Xu, and Rui Zhang. 2019. Energy minimization for wireless communication with rotary-wing UAV. *IEEE Transactions on Wireless Communications* 18, 4 (2019), 2329–2345. https://doi.org/10.1109/twc.2019.2902559

[33] Chongjie Zhang and Victor Lesser. 2013. Coordinating Multi-Agent Reinforcement Learning with Limited Communication. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS '13)*. International Foundation for Autonomous Agents and Multiagent Systems, 1101–1108.

[34] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. 2021. Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. *Handbook of Reinforcement Learning and Control* (2021), 321–384. https://doi.org/10.1007/978-3-030-60990-0_12

[35] Ziyong Zhang, Xiaoling Xu, Jinqiang Cui, and Wei Meng. 2021. Multi-UAV Area Coverage Based on Relative Localization: Algorithms and Optimal UAV Placement. *Sensors* 21 (03 2021), 2400. https://doi.org/10.3390/s21072400

[36] Chenxi Zhao, Junyu Liu, Min Sheng, Wei Teng, Yang Zheng, and Jiandong Li. 2021. Multi-UAV Trajectory Planning for Energy-Efficient Content Coverage: A Decentralized Learning-Based Approach. *IEEE Journal on Selected Areas in Communications* 39, 10 (2021), 3193–3207. https://doi.org/10.1109/JSAC.2021.3088669

[37] Ran Zhuo, Shiqian Song, and Yejun Xu. 2022. UAV communication network modeling and energy consumption optimization based on routing algorithm. *Computational and Mathematical Methods in Medicine* 2022 (2022), 1–10. https://doi.org/10.1155/2022/4782850