Deep Reinforcement Learning Based Three-Dimensional Area Coverage With UAV Swarm

Zhiyu Mou, Yu Zhang, *Graduate Student Member, IEEE*, Feifei Gao, *Fellow, IEEE*, Huangang Wang, Tao Zhang, *Senior Member, IEEE*, and Zhu Han, *Fellow, IEEE*

***Abstract*— Unmanned aerial vehicle (UAV) technology is recognized as a promising solution to area coverage problems (ACPs) and has been extensively studied recently. In this paper, we study the 3D irregular terrain surface coverage problem with a hierarchical UAV swarm. We first build the 3D model of a random irregular terrain and propose a geometric way to project the 3D terrain surface into many weighted 2D patches. Then we develop a two-level hierarchical UAV swarm architecture, including the low-level follower UAVs (FUAVs) and the high-level leader UAVs (LUAVs). For FUAVs, we design a coverage trajectory algorithm to carry out specific coverage tasks within patches based on the star communication topology. For LUAVs, we propose a swarm deep Q-learning (SDQN) reinforcement learning algorithm to select patches. Moreover, an observation history model based on convolutional neural networks (CNNs) and the mean embedding method is integrated into SDQN to address the communication limitation problems of LUAVs. The numerical results show that FUAVs can cover the entire area of each patch with little redundancies , and the total coverage time of the SDQN is less than that of existing methods, which demonstrates the effectiveness of the proposed algorithms.**

***Index Terms*— UAV swarm system, 3D area coverage, deep reinforcement learning, trajectory design.**

Manuscript received October 23, 2020; revised February 24, 2021; accepted April 12, 2021. Date of publication June 14, 2021; date of current version September 16, 2021. The work of Zhiyu Mou, Yu Zhang, and Feifei Gao was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0102401, in part by the National Natural Science Foundation of China under Grant 61831013 and Grant 61771274, and in part by the Beijing Municipal Natural Science Foundation under Grant L182042 and Grant 4212002. The work of Zhu Han was supported in part by the U.S. Multidisciplinary University Research Initiative under Grant 18RT0073 and in part by the NSF under Grant EARS-1839818, Grant CNS1717454, Grant CNS-1731424, and Grant CNS-1702850. *(Corresponding author: Feifei Gao.)*

Zhiyu Mou, Yu Zhang, and Feifei Gao are with the Institute for Artificial Intelligence, Tsinghua University (THUAI), the State Key Laboratory of Intelligent Technologies and Systems, and the Beijing National Research Center for Information Science and Technology (BNRist), Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: mouzy20@mails.tsinghua.edu.cn; z-y16@mails.tsinghua.edu.cn; feifeigao@ieee.org)

Huangang Wang and Tao Zhang are with the Department of Automation, School of Information Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: hgwang@tsinghua.edu.cn; taozhang@tsinghua.edu.cn).

Zhu Han is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 446-701, South Korea (e-mail: zhan2@uh.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2021.3088718.

Digital Object Identifier 10.1109/JSAC.2021.3088718

I. Introduction

AREA coverage problems (ACPs) are important for practical tasks like search and rescue [1], data collection [2], public safety [3], [4], and surveillance [5]. In the early studies, wireless sensor networks (WSNs) [6]–[8] are often adopted for ACPs, which can be viewed as the disk coverage problems. However, WSNs can only be statically deployed in a given area and cannot change their topologies with the terrain. Recently, unmanned aerial vehicle (UAV) technology has been recognized as an alternative solution to the ACPs for its dynamic and flexible properties, and has been studied in the literature [9]–[11]. Many of them assumed the ground area be covered in two-dimensional (2D) environments [9], where the area itself is flat or the UAVs fly high above the ground. Some literature investigated three-dimensional (3D) ACPs in various scenarios. For instance, the authors in [10] studied the 3D space short-time sensing and trajectory planning problem with a single UAV by dividing the 3D space into cuboids. In [11], the authors addressed the 3D UAV deployment problem in uneven terrains for users' coverage and connectivity. The authors in [12] studied the problem of time-optimal 3D urban regular structure coverage with a single UAV using geometrical methods. Moreover, many works have studied the visible coverage problems of 3D non-convex regions using gradient control law based on Voronoi diagrams [13], [14]. Nonetheless, there are many practical scenarios when the coverage has to be carried out on a 3D irregular terrain surface. For example, in the post-disaster searching and rescuing scene, the disaster locations are often in irregular 3D areas. UAVs have to perform data collections or target detections of the whole terrain surface. It is then vital and necessary to study the ACPs of 3D terrain surfaces with UAV technology.

For large-scale 3D areas, a single small UAV cannot accomplish the complex mission by itself due to the limited working range and energy capacity. A single large UAV may have more coverage capacities but is not energy-efficient and is not easy-controlled [15], [16]. It is then found that the UAV swarm that contains a large number of low-cost UAVs can efficiently complete the ACPs in large-scale scenarios [17].

The very original swarm algorithms are inspired by the biology behavior of insect communication, such as ant colony optimization (ACO) [18], particle swarm optimization (PSO) [19],

and artificial bee colony (ABC) algorithm [20]. However, most bio-inspired algorithms are intuitive methods, which cannot achieve satisfactory performance when being transplanted to the UAV aided environment coverage.

Recently, researchers attempt to use deep reinforcement learning (DRL) algorithms in UAV swarm, especially the multi-agent DRL algorithms that have solid theoretical foundations. There are two main challenges in applying multi-agent DRL algorithms to UAV swarm [21]. The first challenge lies in the high dimensions of states and observations, as the number of individuals in UAV swarm is very large. High dimensional observations can decrease the reinforcement learning (RL) algorithms' efficiencies of searching optimal policies in observation-action space. The second challenge is that individuals of UAV swarm can only obtain the information from several UAVs due to the communication range constraints, which makes the entire environment partially observed by each UAV. Some researchers have made some progress in addressing these challenges in shortest path problems [22], and in several classical swarm problems [21]. However, the challenges of applying multi-agent DRL to UAV swarm remain unsolved in 3D terrain surface coverage problems.

In this paper, we study the 3D irregular terrain surface coverage problem with UAV swarm based on multi-agent DRL algorithm. The contributions are summarized in three aspects. Firstly, we design a hierarchical UAV swarm architecture, including follower UAVs (FUAVs) and leader UAVs (LUAVs) to improve the learning efficiency and reduce the communication overloads. Secondly, we build the 3D model of a random irregular terrain and develop a geometrical way to project the 3D irregular terrain surface into many weighted *patches*. Then we design a trajectory planning scheme for FUAVs to carry out specific coverages within patches, and propose a swarm deep Q-learning (SDQN) reinforcement algorithm for LUAVs to select patches. Moreover, an observation history model based on convolutional neural networks (CNNs) and mean embedding method are integrated into SDQN to address the communication limitation problems of LUAVs. Simulation results show that FUAVs can cover the entire area of each patch with little redundancies, and the total coverage time of SDQN is less than that of existing methods, which demonstrates the effectiveness of the proposed algorithms.

The rest parts of this paper are organized as follows. Section II presents the system models of the ACP. Section III proposes the calculation method for patch areas and a trajectory design algorithm for FUAVs. Section IV focuses on the analysis of LUAVs' patch selection. In Section V, we develop the SDQN algorithm to solve the patch selection problem of LUAVs. Simulation results and analysis are provided in Section VI, and conclusions are made in Section VII.

Notations: x, $\mathbf{x}$, $\mathbf{X}$ represent a scalar x, a vector $\mathbf{x}$ and a matrix $\mathbf{X}$, respectively; $\sum$, $\mathbb{E}(\cdot)$, $\max$ and ∇ denote the sum, expectation, maximum and vector differential operation, respectively; $\lceil\cdot\rceil$ denotes the ceiling function; $(\cdot)^T$ denotes the transpose of a matrix; $vec(\cdot)$ denotes the column vectorization function of the matrix; $\oplus$ and $\circ$ denote the matrix or operation and the logical or operand, respectively; Besides, $\triangle ABC$ denotes the triangle formed by point A, B and C; S_{ABC} denotes the area of $\triangle ABC$; $\overset{\frown}{AB}$ and $\overline{AB}$ denote the curved line and the straight connection line between points A and B, respectively.

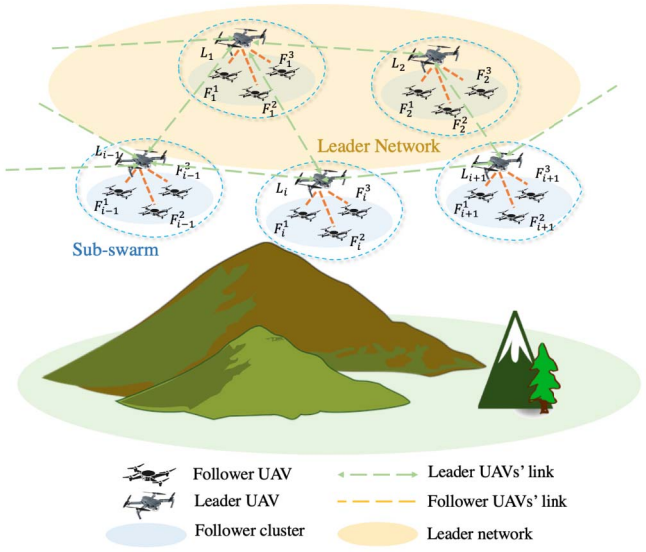


Fig. 1. 3D terrain area coverage using UAV swarm.

II. System Model

We consider a UAV swarm assisted 3D irregular terrain surface coverage scenario, where the whole UAV swarm is divided into N sub-swarms and sub-swarm i is composed of one LUAV L_i and M_i FUAVs $F_{i,j}$, $i \in \mathcal{N} = \{1, 2, \ldots, N\}$, $j \in \{1, 2, \ldots, M_i\}$, as shown in Fig. 1. Denote the 3D terrain to be covered as $\mathcal{Q}$. Without loss of generality, we set M_i the same for all sub-swarms, i.e., $M_i = M$, $\forall i \in \mathcal{N}$, and define $\mathcal{M} = \{1, 2, \ldots, M\}$. The communication topology in each sub-swarm adopts the star network,[1] where LUAV acts as the communication center and can directly communicate with all FUAVs by constant power $P_{T,L}$. The FUAVs are designed to only communicate to their LUAVs by constant power $P_{T,F}$ but with no connections to other FUAVs. Meanwhile, LUAVs will formulate a *leader communication network* (LCN), where LUAVs can transmit signals to other LUAVs with constant power $P_{T,L}$. Two distinct LUAVs in the LCN can establish a *communication link* (CL) to communicate with each other directly if the received signal power [2] exceeds a threshold P_0. When carrying out a task, LUAVs decide the parts of $\mathcal{Q}$ to be covered and lead the FUAVs to complete the area coverage.

A. 3D Digital Elevation Model

An irregular 3D terrain $\mathcal{Q}$ is shown in Fig. 2 , where the shape of the top view $\mathcal{Q}$ on the plane is a rectangle with width L_x and length L_y. A 3D X-Y-Z Cartesian coordinate is set

[1]Note that many other communication topologies, such as FANET, star topology of multiple FANETs, FANET of UAV clusters topology, etc., can be applied to the sub-swarm system. We adopt the star topology in the sub-swarm since the star topology has the advantages of global supervisions, centralized control and high communication efficiencies [23].

[2]Note that the received signal powers of two LUAVs are the same due to symmetry

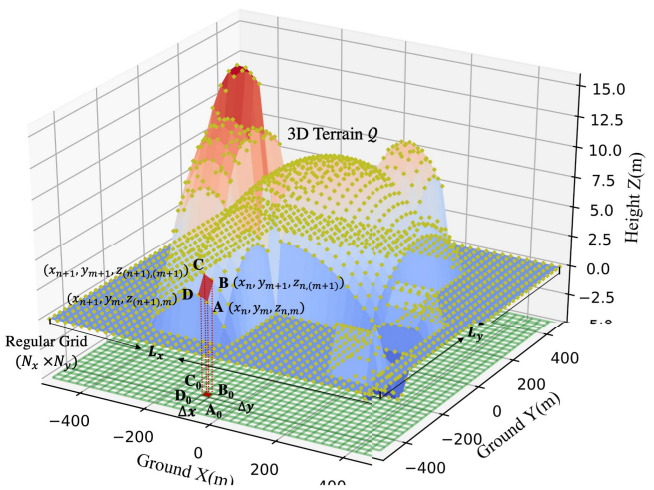


Fig. 2. The DEM of the 3D irregular terrain $\mathcal{Q}$, where yellow points on $\mathcal{Q}$ represent the sampling points, and the green grid below $\mathcal{Q}$ is the regular grid.

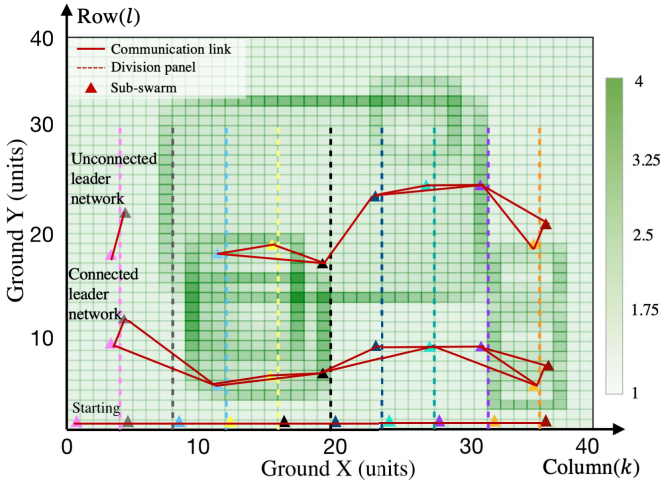


Fig. 3. 2D hotmap of terrain $\mathcal{Q}$, and the lanes assigned for sub-swarms for UAV swarms.

for $\mathcal{Q}$ where the X-Y plane represents the ground. We adopt the *regular grid method* to do the sampling of terrain $\mathcal{Q}$. The regular grid method samples the elevations of points on $\mathcal{Q}$ using a regular grid and builds the terrain model based on the collected elevation data. The bottom grid in Fig. 2 represents the regular grid, while the scattered points on $\mathcal{Q}$ represent the sampling points. Specifically, the regular grid is an $L_x \times L_y$ rectangle formed by small rectangles with width Δx and length Δy. The number of rectangles is $N_x \times N_y$, where

$$N_x = \left\lceil \frac{L_x}{\Delta x} \right\rceil, \quad N_y = \left\lceil \frac{L_y}{\Delta y} \right\rceil. \tag{1}$$

Note that Δx and Δy are hyperparameters that can be set manually. As the values of Δx and Δy become smaller, we can obtain a more accurate model of $\mathcal{Q}$, but need to measure larger amount of data. The regular grid is placed under the terrain $\mathcal{Q}$ and coincides with the projection of $\mathcal{Q}$ on the ground. The sampling points on $\mathcal{Q}$ and vertices of the regular grid are in one-to-one correspondence. The number of sampling points is $(N_x+1)\times(N_y+1)$ that is the same as the number of vertices. The coordinate of sampling point in the X-Y-Z coordinate system is denoted as $(x_n, y_m, z_{n,m})$ that satisfies

$$|x_n - x_{n+1}| = \Delta x, \quad |y_m - y_{m+1}| = \Delta y, \tag{2}$$

where $n \in \{1, 2, \ldots, N_x + 1\}, m \in \{1, 2, \ldots, N_y + 1\}$. Note that x_n and y_m of the sampling points are known in advance, while $z_{n,m}$'s are unknown and needs to be measured. According to the coordinates of sampling points, we can approximately build the model of $\mathcal{Q}$ using the interpolation method [24]. The model of $\mathcal{Q}$ is named as the *digital elevation model* (DEM) in the realm of surveying and mapping engineering [25].

Moreover, each rectangle in the regular grid corresponds to a *patch* of $\mathcal{Q}$ that is a smooth 3D surface.The patches of $\mathcal{Q}$ and the rectangles of the regular grid are also in one-to-one correspondence. In Fig. 2, $A_0B_0C_0D_0$ is a rectangle of the regular grid, and $ABCD$ is the corresponding patch of $\mathcal{Q}$, where A, B, C, D are four sampling points. We suppose the coordinates of points A, B, C, and D are $(x_n, y_m, z_{n,m})$, $(x_n, y_{m+1}, z_{n,(m+1)})$, $(x_{n+1}, y_{m+1}, z_{(n+1),(m+1)})$ and $(x_{n+1}, y_m, z_{(n+1),m})$, respectively.

The sub-swarms cover $\mathcal{Q}$ patch by patch and the covering time of each patch is related to the patch area size. We need to calculate the area of each patch for UAV swarm's coverage time computing. The detailed area calculation algorithm of patches are presented in Section III-A. The information of patch areas can be expressed in a *hotmap* that is a regular grid formed by rectangles of different color depths, as shown in Fig. 3. The rectangle in column k and row l is denoted as rectangle (k, l), and its corresponding patch of $\mathcal{Q}$ is denoted as patch (k, l), $k \in \{1, 2, \ldots, N_x\}$ and $l \in \{1, 2, \ldots, N_y\}$. The color depth d_{kl} of each rectangle (k, l) is determined by

$$d_{kl} = \frac{S_{kl}}{\Delta x \Delta y} \geq 1, \tag{3}$$

where S_{kl} represents the area of patch (k, l), and the area of rectangle (k, l) is $\Delta x \Delta y$. Note that the minimum area of all the patches is the area of the rectangles $\Delta x \Delta y$, which makes d_{kl} not less than 1. The information of the hotmap of terrain $\mathcal{Q}$ can be represented in an $N_x \times N_y$ matrix $\mathbf{S}_p = (S_{kl})$, $k \in \{1, 2, \ldots, N_x\}$ and $l \in \{1, 2, \ldots, N_y\}$.

B. Communication Link Model

We model the communication channels between LUAVs in the LCN as *air-to-air* (A2A) communication links [26]. Denote the position of LUAV L_i as $\mathbf{p}_i$. Then the LUAV L_i's received power of signals transmitting from LUAV L_j, $P_r(\mathbf{p}_i, \mathbf{p}_j)$, can be written as[3]

$$P_r(\mathbf{p}_i, \mathbf{p}_j) = P_{T,L} + G_r + G_t - P_l(\mathbf{p}_i, \mathbf{p}_j), \tag{4}$$

where G_r and G_t are the constant antenna gains of LUAV L_i and L_j, respectively, and $P_l(i, j)$ is the large-scale fading effect. As there is no ground obstacle between LUAVs,

[3]Note that the units of the variables in (4) are all dBs.

the large-scale $P_l(\mathbf{p}_i, \mathbf{p}_j)$ can be expressed as

$$P_l(\mathbf{p}_i, \mathbf{p}_j) = 10\alpha \log_{10}\left(\frac{4\pi \left\|\mathbf{p}_i - \mathbf{p}_j\right\|_2 f_c}{v_c}\right), \tag{5}$$

where $\alpha > 0$ is the path loss exponent, f_c is the electromagnetic wave frequency, and v_c is the speed of light. Hence, any two LUAVs L_i and L_j can establish a CL E_{ij}, if $P_r(\mathbf{p}_i, \mathbf{p}_j) \geq P_0$, i.e., the distance between two LUAVs satisfies

$$\left\|\mathbf{p}_i - \mathbf{p}_j\right\|_2 \leq \frac{v_c}{4\pi f_c} \exp\left\{\frac{\ln(10)}{10\alpha}(P_{T,L} + G_r + G_t - P_0)\right\}, \tag{6}$$

and LUAVs L_i and L_j cannot communicate with each other directly otherwise.

Similarly, the communication channels between LUAVs and FUAVs within sub-swarms are also modeled as A2A links. Denote the position of FUAV $F_{i,j}$ as $\mathbf{p}_{i,j}$. Then the LUAV L_i's received power of signals transmitting from FUAV $F_{i,j}$, $P_{r,L}(\mathbf{p}_i, \mathbf{p}_{i,j})$, can be written as

$$P_{r,L}(\mathbf{p}_i, \mathbf{p}_{i,j}) = P_{T,F} + G_r + G_t - P_l(\mathbf{p}_i, \mathbf{p}_{i,j}), \tag{7}$$

and the FUAV $F_{i,j}$'s received power of signals transmitting from LUAV L_i, $P_{r,F}(\mathbf{p}_{i,j}, \mathbf{p}_i)$, can be written as

$$P_{r,L}(\mathbf{p}_{i,j}, \mathbf{p}_i) = P_{T,L} + G_r + G_t - P_l(\mathbf{p}_{i,j}, \mathbf{p}_i). \tag{8}$$

To make sure that LUAVs and their FUAVs can communicate with each other directly within patches, the size of any patch $ABCD$ should satisfy

$$\max\{|\overline{AC}|, |\overline{BD}|\} \leq \frac{v_c}{4\pi f_c} \exp\left\{\frac{\ln(10)}{10\alpha}\left[\min\{P_{T,L}, P_{T,F}\} + G_r + G_t - P_0\right]\right\}, \tag{9}$$

where $|\overline{AC}|$ and $|\overline{BD}|$ can be calculated using the proposed algorithm in Section III-A.

C. Leader Communication Network Structure

The LCN can be viewed as a weighted undirected graph $\mathcal{G} = \{\mathcal{L}, \mathcal{E}\}$, where $\mathcal{L}$ represents the set of all LUAVs in the UAV swarm, i.e., $\mathcal{L} = \{L_i | i \in \mathcal{N}\}$, and $\mathcal{E}$ represents the set of all CLs between LUAVs, i.e., $\mathcal{E} = \{E_{ij} | i, j \in \mathcal{N}, i \neq j\}$. LUAVs that have CLs with LUAV L_i are called the *neighbors* of L_i. All the neighbors of LUAV L_i form the neighbor set $\mathcal{N}(i)$, i.e., $\mathcal{N}_i = \{L_{i'} | E_{ii'} \in \mathcal{E}, i \neq i'\}$. Denote the number of neighbors of LUAV L_i as n_i, i.e., $|\mathcal{N}_i| = n_i$. The information of neighbor sets of LUAVs can be represented in a connectivity matrix $\mathbf{V} = (v_{ii'}) \in \mathbb{R}^{N \times N}$, where $v_{ii'} = 1$ if the CL exists between LUAV L_i and $L_{i'}$, and $v_{ii'} = 0$ otherwise. Moreover, LUAVs are constrained to obtain information only from their neighbors.

A *walk* in graph $\mathcal{G}$ is a sequence that is composed of LUAVs and CLs alternately. It can be defined as

$$\mathcal{W} := L_1 E_{12} L_2 E_{23} \ldots L_{w-1} E_{(w-1)w} L_w, \tag{10}$$

where LUAV L_{i+1} and L_i are the two endpoints of the CL $E_{i(i+1)}$, respectively, $i \in \{1, 2, \ldots, w-1\}$ and $w \geq 2$. In fact $\mathcal{W}$ can be regarded as a communication path from the first LUAV L_1 to the last one L_n. If the messages of any LUAV L_i can reach any other LUAVs $L_{i'}$ through a walk, $i, i' \in \mathcal{N}$, we say that the LCN $\mathcal{G}$ is a connected graph; otherwise, $\mathcal{G}$ is a disconnected graph. Examples of the connected and disconnect LCN are shown in Fig. 3, where the solid lines represent the CLs of LUAVs. Denote a binary variable $f_{c,t}$ as the connection indicator of the LCN $\mathcal{G}$ at time step t, i.e., $f_{c,t} = 1$ if $\mathcal{G}$ is connected at time step t, while $f_{c,t} = 0$ otherwise. During the coverage task, the LCN is required to maintain a connected graph, i.e., $f_{c,t} = 1, \ \forall t$.

D. Time Consumptions and Coverage Rates

As shown in Fig. 3, each triangle represents a sub-swarm of UAVs. All the sub-swarms start on the patches along the X-axis. The starting patch of sub-swarm i is denoted as $p_{i,0}$, $i \in \mathcal{N}$. The distance d between the starting points of any two adjacent sub-swarms is the same, i.e., $d = \frac{L_x}{N}$. The FUAVs of each sub-swarm i first cover their starting patch $p_{i,0}$ under the coordination of LUAV L_i, while LUAV L_i selects the next patch $p_{i,1}$ to be covered. During the coverage of patch $p_{i,0}$, FUAVs fly continuously and carry out detailed coverages, such as collections or detections, during the flight. When finishing covering patch $p_{i,0}$, sub-swarm i fly to the next patch $p_{i,1}$ for coverage. With this pattern, once the FUAVs of sub-swarm i finish covering their q-th ($q \geq 0$) patch $p_{i,q}$, the whole sub-swarm transfers to the next patch $p_{i,(q+1)}$. Each sub-swarm i covers the patches one after another until the whole terrain $\mathcal{Q}$ is covered.

1) Time Consumptions: Denote Q_i as the total number of patches covered by sub-swarm i. For each sub-swarm i, the total time consumed in the coverage task $T_{f,i}$ is the summation of two parts, including: the *collection time of FUAVs* when carrying out specific coverage in patches, and the *patrolling time of the sub-swarm* when transferring between different patches.

a) Collection time of FUAVs: The collection time is the time when FUAVs fly continuously within patches and carry out specific coverages, such as data collections or target detections. The collection time per meter is set to be a constant C_{fc}. Denote the flying distance of FUAV F_i^j, $j \in \mathcal{M}$ in the q-th patch as $U_{i,q}^j$. As all the FUAVs fly concurrently, the total time of sub-swarm i for covering the q-th patch $\xi_{i,q}$ can be expressed as

$$\xi_{i,q} = \max_{j \in \mathcal{M}}\{C_{fc} U_{i,q}^j\} = C_{fc} \max_{j \in \mathcal{M}}\{U_{i,q}^j\}. \tag{11}$$

b) Patrolling time of sub-swarms: When transferring between the q-th patch and $(q+1)$-th patch, the patrolling time of sub-swarm i is denoted as $C_{q,(q+1)}$. The total patrolling time of sub-swarm i can be expressed as

$$\eta_i = \sum_{q=1}^{Q_i - 1} C_{q,(q+1)}. \tag{12}$$

c) Coverage time for sub-swarm i: The total coverage time of sub-swarm i is the summation of $\xi_{i,q}$ in all Q_i patches

and η_i, i.e.,

$$T_{f,i} = \sum_{q=1}^{Q_i} \xi_{i,q} + \eta_i. \tag{13}$$

d) Overall coverage time: As all the sub-swarms cover the terrain surface concurrently, the overall coverage time T_f is the maximum coverage time among all the sub-swarms, i.e.,

$$\begin{aligned} T_f &= \max_{i\in\mathcal{N}}\{T_{f,i}\} \\ &= \max_{i\in\mathcal{N}}\{\sum_{q=1}^{Q_i} C_{fc} \max_{j\in\mathcal{M}}\{U_{i,q}^j\} + \sum_{q=1}^{Q_i-1} C_{q,(q+1)}\}. \end{aligned} \tag{14}$$

Note that as the collection time of different patches vary, sub-swarms may finish covering the current patches asynchronously. The total coverage time is restricted in T_0 time steps. However, UAV swarm can end the task early if the swarm covers all the patches within T_0 time steps, i.e., $T_f < T_0$. Hence, the actual terminate time of the coverage task T can be expressed as $T = \min\{T_f, T_0\}$.

2) Coverage Rates: To represent the coverage situations of patches during the mission, we define an $N_x \times N_y$ *coverage matrix* $\mathbf{C} = (c_{kl})$, $c_{kl} \in \{0, 1\}$, where $c_{kl} = 0$ denotes that the patch (k, l) is not covered by any UAV sub-swarms while $c_{kl} = 1$ represents that the patch (k, l) has already been covered by at least one sub-swarm. We also define the *individual coverage matrix* of sub-swarm i as $\mathbf{C}_i = (c_{kl}^i) \in \mathbb{R}^{N_x \times N_y}$, where $c_{kl}^i = 1$ means that the patch (k, l) has already been covered by sub-swarm i while $c_{kl}^i = 0$ means the opposite.

a) Overall coverage rate: The overall coverage rate r_a^c is defined as the ratio between the area of covered patches and the area of the entire terrain, i.e.,

$$r_a^c = \frac{vec(\mathbf{C})^T vec(\mathbf{S}_p)}{\mathbf{1}^T vec(\mathbf{S}_p)}, \tag{15}$$

where $\mathbf{1}^T$ represents a $1 \times N_x N_y$ vector with all 1 components. Note that r_a^c is in the range of $[0, 1]$.

b) Individual coverage rate: The individual coverage rate of sub-swarm i can be defined as

$$r_i^c = \frac{vec(\mathbf{C}_i)^T vec(\mathbf{S}_p)}{\mathbf{1}^T vec(\mathbf{S}_p)}. \tag{16}$$

Note that r_i^c belongs to the set $[0, 1]$, $\forall i \in \mathcal{N}$.

c) Repeated coverage rate: As different sub-swarms may repeatedly cover the same patches, the overall coverage rate r_a^c cannot be more than the summation of the individual coverage rates of all sub-swarms, i.e., $\sum_{i=1}^N r_i^c \geq r_a^c$. The repeated coverage rate r_p^c is defined as

$$r_p^c = \frac{1}{N}\left[\left(\sum_{i=1}^N r_i^c\right) - r_a^c\right]. \tag{17}$$

Note that the repeated coverage rate is always non-negative, i.e., $r_p^c \geq 0$.

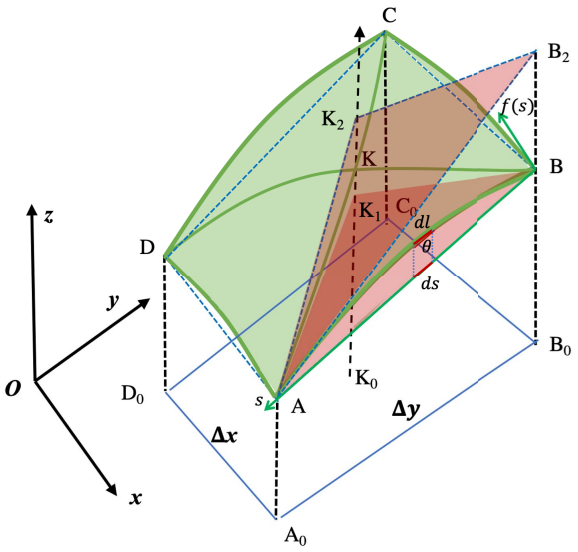


Fig. 4. Geometrical relationship between patch $ABCD$ and rectangle $A_0B_0C_0D_0$.

E. Problem Formulation

The goal of the ACP is twofold. On the one hand, the UAV swarm should try to cover the given terrain $\mathcal{Q}$ as much as possible, or equivalently maximizing r_a^c during the process of coverage. On the other hand, the actual terminate time T needs to be minimized. The whole coverage problem can be expressed as an optimization problem, i.e.,

$$\max_{T,Q_1,Q_2,\ldots,Q_N} -\beta_1 T + \beta_2 \sum_{t=0}^{T_f} r_{a,t}^c \tag{18}$$

$$\text{s. t. } r_p^c \leq \theta_c, \tag{18a}$$

$$f_{c,t} = 1, \quad \forall t \in [0, T], \tag{18b}$$

where (18a) indicates that the repeated coverage rate r_p^c should not exceed a threshold θ_c, i.e., $r_p^c \leq \theta_c$, since large r_p^c implies low coverage efficiency, and (18b) represents that the LCN $\mathcal{G}$ should maintain a connected graph during the flight. In the following, we propose an efficient coverage algorithm including two efficient algorithms for the coverage trajectory design of FUAVs and patch selections of LUAVs, respectively.

III. Patch Area Calculation and Trajectory Design of FUAVs

As the coverage time of patches of FUAVs is related to the area size of the patches, the calculation of patch areas is a necessity for solving the coverage problem. Moreover, we should design the trajectory for FUAVs to cover each single patch.

A. Patch Area Calculation Algorithm

Inspired from [27], we here propose a patch area calculation method that separates each patch into curved triangles and sum up their approximated areas as the entire area of the patch. Specifically, as shown in Fig. 4, we divide patch $ABCD$ into four curved triangles, i.e., $\triangle ABK$, $\triangle CBK$, $\triangle CDK$, and $\triangle ADK$, where point K on $ABCD$ is right

above $A_0B_0C_0D_0$'s center K_0. Although we cannot compute the accurate areas of these triangles, we can obtain the upper bound and lower bound of their areas. For curved $\triangle ABK$, its minimum area is the triangle area formed by the straight lines $\overline{AB}$, $\overline{BK_1}$ and $\overline{AK_1}$, where K_1 is at the center of the connection line $\overline{AC}$. The length of $\overline{AB}$ can be calculated as

$$|\overline{AB}| = \sqrt{(\Delta y)^2 + (z_{n,(m+1)} - z_{n,m})^2}, \tag{19}$$

and the length of $\overline{AK_1}$ and $\overline{BK_1}$ are both

$$|\overline{AK_1}| = |\overline{BK_1}| = \frac{\sqrt{(\Delta x)^2 + (\Delta y)^2}}{2}. \tag{20}$$

The lower bound of the curved $\triangle ABK$ area can be calculated by the Helen formula [28], i.e.,

$$\begin{aligned} S_{ABK} &\geq S_{ABK_1} \\ &= \sqrt{p(p - |\overline{AB}|)(p - |\overline{AK_1}|)(p - |\overline{BK_1}|)}, \end{aligned} \tag{21}$$

where p is half of the perimeter of $\triangle ABK_1$, i.e.,

$$p = \frac{|\overline{AB}| + |\overline{AK_1}| + |\overline{BK_1}|}{2}. \tag{22}$$

The maximal area of curved $\triangle ABK$ is the area of $\triangle AB_2K_2$ formed by $\overline{AB_2}$, $\overline{AK_2}$, and $\overline{K_2B_2}$ whose lengths are the upper bounds of the lengths of $\overset{\frown}{AB}$, $\overset{\frown}{AK}$, and $\overset{\frown}{BK}$, respectively. Denote the *maximum relative slope* of each point on $\mathcal{Q}$ in all directions as $C_{ms} < \infty$. It can be proved that $|\overline{AB_2}|$, $|\overline{AK_2}|$, and $|\overline{B_2K_2}|$ are:

$$|\overline{AB_2}| = \sqrt{\frac{(\Delta y)^2 + (z_{n,m} - z_{n,(m+1)})^2}{1 + {C_{ms}}^2}}, \tag{23}$$

$$|\overline{AK_2}| = \sqrt{\frac{(\Delta y)^2 + (\Delta x)^2 + (z_{n,m} - z_{(n+1),(m+1)})^2}{1 + {C_{ms}}^2}}, \tag{24}$$

$$|\overline{B_2K_2}| = \sqrt{\frac{(\Delta y)^2 + (\Delta x)^2 + (z_{(n+1),m} - z_{n,(m+1)})^2}{1 + {C_{ms}}^2}}. \tag{25}$$

The definition of C_{ms} and derivations of the length of $|\overline{AB_2}|$, $|\overline{AK_2}|$, and $|\overline{B_2K_2}|$ are shown in Appendix. Hence, the upper bound of the curved $\triangle ABK$ area can be expressed as

$$\begin{aligned} S_{ABK} &\leq S_{AB_2K_2} \\ &= \sqrt{p'(p' - |\overline{AB_2}|)(p' - |\overline{AK_2}|)(p' - |\overline{B_2K_2}|)}, \end{aligned} \tag{26}$$

where p' is half of the perimeter of $\triangle AB_2K_2$, i.e.,

$$p' = \frac{|\overline{AB_2}| + |\overline{AK_2}| + |\overline{B_2K_2}|}{2}. \tag{27}$$

The area of curved $\triangle ABK$ can be approximated by a convex combination of the lower bound and upper bound of S_{ABK}, i.e.,

$$S_{ABK} \approx \alpha S_{ABK_1} + (1 - \alpha) S_{AB_2K_2}, \tag{28}$$

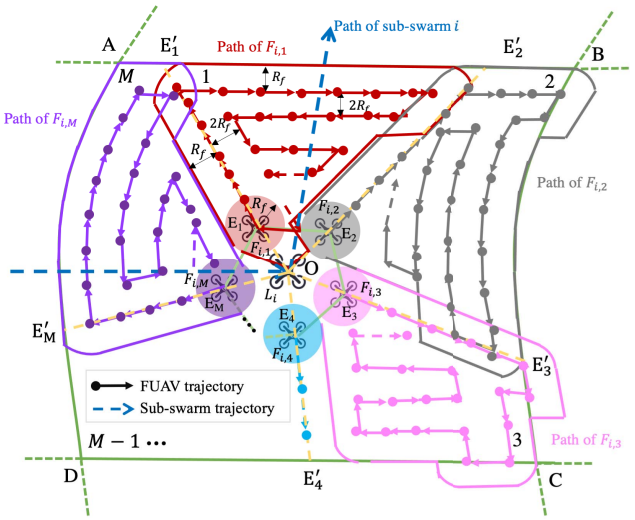


Fig. 5. Single patch coverage trajectories of FUAVs. The patch is divided into M cells for M FUAVs to cover using spiral-zigzag search pattern. The coverage range of FUAV is a circle with radius R_f. FUAVs fly along the trajectories continuously.

where $\alpha \in (0, 1)$ is the weight of the lower bound that can be set manually. The area of curved $\triangle ADK$, $\triangle CDK$, and $\triangle BCK$ can be calculated in the same way of curved $\triangle ABK$. Hence, the area of patch $ABCD$ can be derived as

$$S_{ABCD} = S_{ABK} + S_{BCK} + S_{CDK} + S_{ADK}, \tag{29}$$

where S_{ABCD} represents the area of patch $ABCD$. The areas of other patches of $\mathcal{Q}$ can be calculated in the same manner.

B. FUAVs Coverage Trajectory Design

We assume the FUAVs can adjust their vertical speeds automatically according to the topography to maintain a constant height H above the terrain surface. The coverage range of each FUAV on the patch is a circle with constant radius R_f, which indicates that the coverage abilities of all the FUAVs are the same. Since the size of each patch is relatively small compared to the size of the whole terrain ($\Delta x \ll L_x, \Delta y \ll L_y$) and the relative slope of $\mathcal{Q}$ is constrained within a certain range, each patch can be viewed as a flat 3D surface.

The trajectory design of FUAVs contains two stages: cells decomposition and searching trajectory [29]–[31]. Fig. 5 shows the overall trajectory designs of covering a random patch $ABCD$ for FUAVs of sub-swarm i. At the beginning, the FUAVs surround LUAV L_i forming a regular M-sided polygonal formation $E_1E_2 \ldots E_M$, where the FUAV $F_{i,j}$ is at vertex E_j and LUAV L_i is at the center O. The angles between any two adjacent FUAVs are the same, i.e.,

$$\angle E_1OE_2 = \angle E_2OE_3 = \ldots = \angle E_MOE_1 = \frac{360^\circ}{M}. \tag{30}$$

The distance between LUAV L_i and any FUAV $F_{i,j}$ equals to the coverage radius of FUAVs. In the first stage, we decompose patch $ABCD$ into M cells by extending the segment lines OE_j to OE'_j, $j \in \mathcal{M}$. The areas of all the cells are nearly the same and each cell j is assigned to one FUAV $F_{i,j}$. Each FUAV covers its own cell based on the searching trajectories designed in the second stage. The searching trajectory patterns

are various, including spiral pattern, lawnmower pattern, and zamboni pattern, etc [29]. We combine the spiral pattern with Zigzag pattern and propose a *spiral-Zigzag* searching pattern for the coverage trajectory design of FUAVs. Specifically, each FUAV $F_{i,j}$ first flies along its cell boundaries, including OE'_j and the boundaries of patch $ABCD$ except for the last boundary of its cell. Note that the FUAV turns to the direction of the next boundary before flying to the place right above it. The distance between the FUAV $F_{i,j}$ and the boundaries except for OE'_j is the radius of the coverage circle R_f. Secondly, FUAV $F_{i,j}$ moves right above the last boundary of cell and flies for a certain distance along OE'_{j+1}. Then the FUAV $F_{i,j}$ covers the remaining cell areas based on the Zigzag path as shown in Fig. 5 and flies back to the starting point E_j in the end. The distances between adjacent coverage patches in Zigzag trajectory are all $2R_f$. Other patches of $\mathcal{Q}$ are covered in the same way.

The coverage times of all cells are nearly the same due to the same coverage capacities of FUAVs. The FUAVs of sub-swarm i are designed to cover the cells concurrently. Hence, the total coverage time of each patch is nearly the coverage time of a cell and is proportional to the area of the patch. As FUAVs have to receive commands from LUAVs and upload collected data to LUAVs, the communications between LUAVs and FUAVs happen all the time during the coverage task. The total communication overload of each FUAV is $\mathcal{O}(T)$, since FUAVs only communicate with their LUAVs. As each LUAV communicates with M FUAVs simultaneously, the total communication overload of each LUAV is $\mathcal{O}(MT)$.

IV. Terrain Panel Divisions and LCN Connectivity Examination

To reduce the difficulties for LUAVs to select patch, we propose a terrain division method to divide $\mathcal{Q}$ into several lanes based on the prior knowledge. Besides, we present an examination method to check the connectivity of LCN.

A. Terrain Panel Divisions

Before the discussion of the patch selection algorithm for LUAVs, we plan the coverage range for each sub-swarm in advance based on the features of the terrain and the structures of the UAV swam. As all the sub-swarms start from the same side of the terrain $\mathcal{Q}$ separated by identical distance, we assign each sub-swarm i to a straight lane with width w_i, $i \in \mathcal{N}$. The widths of all the lanes are designed to be the same, i.e.,

$$w_1 = w_2 = \dots w_N = \frac{N_x}{N}\Delta x = \left\lceil \frac{L_x}{\Delta x} \right\rceil \frac{\Delta x}{N}. \tag{31}$$

The lanes are separated by impenetrable panels, which indicates that LUAVs cannot select patches across panels. The panels and lanes of sub-swarms are shown in Fig. 3, where dotted lines represent the panels between lanes. The length of the panel for sub-swarm i is denoted as l_i and should satisfy $0 \le l_i \le \left\lceil \frac{L_y}{\Delta y} \right\rceil \Delta y$, $i \in \{1, 2, \dots, N-1\}$. Here, we set the length of all the panels to be the same, i.e.,

$$l_1 = l_2 = \dots = l_{N-1} = \alpha_l \left\lceil \frac{L_y}{\Delta y} \right\rceil \Delta y, \tag{32}$$

where $\alpha_l \in [0, 1]$ can be set manually.

The panel divisions of terrain allow the sub-swarms to cover the patches of their own lanes in priority, which can avoid low coverage efficiency and chaotic swarm behaviors caused by unconstrained patch selections of LUAVs. At the same time, the panel divisions can enable the sub-swarms that have already completed the coverage mission in their own lanes to help other UAVs that have not finished.

B. Connectivity Examination of LCN

We adopt the *depth first search* (DFS) [32] algorithm to check the dynamic connectivity of $\mathcal{G}$ at each time step during the coverage. The details of weighted quick-union algorithm are presented in Algorithm 1. The overall approximate time complexity of DFS is $\mathcal{O}(N^2)$. Note that $S[-1]$ represents the top element of stack S.

Algorithm 1 Examining the Connectivity of $\mathcal{G}$ Using Depth First Search

Inputs: The LCN $\mathcal{G}$, the connectivity matrix $\mathbf{V}$, and the number of LUAVs N.
Outputs: Whether $\mathcal{G}$ is connected or not (True or False).
Initialize: A LUAV stack S, an initial root LUAV $L_0 \in \mathcal{G}$, a counter $Count = 0$, current LUAV L_{temp}, a visited list a with N zeros, and a flag $f = False$.

```
Visit LUAV l_0, i.e., a[0] = 1
Push l_0 into S, S[-1] = l_0;
Increase the counter Count ← Count + 1;
while the length of S is not zero do:
    Make the current LUAV be the top LUAV of the stack S, i.e. l_temp = S[-1].
    Make flag to be True, f = True;
    for i = 1 to N do
        if V[l_temp, i] = 1 and a[i] = 0 then
            Visit LUAV l_i, i.e. a[i] = 1;
            Push l_i into S, S[-1] = l_i;
            Increase the counter Count ← Count + 1, and make flag to be False, f = False;
            Break;
        end if
    end for
    if f = True then
        Pop the top LUAV from stack S;
    end if
end while
if Count = N then
    Return True;
else
    Return False.
end if
```

V. Patch Selections of LUAVs Using Swarm Deep Q-Network

The LUAV patch selection problem can be modeled as a Markov stochastic process [9], [33], [34]. Based on characteristics of the LCN and deep Q-learning algorithm, we propose

a swarm deep Q-learning (SDQN) algorithm to solve the Markov model of LUAVs' patch selection problem.

A. Decentralized Partially Observed Markov Decision Process

During the area coverage mission, LUAVs are responsible for selecting the new target patches to cover and leading FUAVs while flying between different patches. All LUAVs should cooperate to achieve the goals of the coverage problem. Note that LUAVs can only communicate with their neighbors, which indicates that LUAVs are locally observed when making decisions on patch selection. Hence, the patch selection problem of LUAVs can be modeled as a *decentralized partially observed Markov decision process* (Dec-POMDP) that is defined below.

Definition 1 (Dec-POMDP): A decentralized partially observed Markov decision process is defined by a tuple $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}_i\}_{i\in\mathcal{N}}, \{\Omega_i\}_{i\in\mathcal{N}}, \mathcal{O}, \mathcal{P}, R, \gamma)$, *where* $\mathcal{N} = \{1, 2, \dots N\}$ *denotes the set of* $N(N > 1)$ *agents;* $\mathcal{S}$ *is the joint state space of all agents;* $\mathcal{A}_i$ *is the action space of agent* i*;* Ω_i *is the observation space of agent* i*, and* $\Omega = \{\Omega_i\}_{i\in\mathcal{N}}$ *is the set of joint observations;* $\mathcal{O} : \mathcal{S} \times \mathcal{N} \to \Delta(\Omega)$ *is the probability function of observations given states for any agent;* $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ *is the state transition probability function from any state* $\mathbf{s} \in \mathcal{S}$ *to other state* $\mathbf{s}' \in \mathcal{S}$ *for any joint action* $\mathbf{a} \in \mathcal{A}$*, where* $\mathcal{A} \triangleq \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_N$ *is the joint action space of all agents;* $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ *is the common immediate reward function of all agents, and* $\gamma \in [0, 1]$ *is a discounted factor.*

The details of the basic elements in Dec-POMDP of patch selection problem are described as follows.

Agents: Each LUAV can be viewed as an agent in Dec-POMDP. The UAV swarm covers the terrain $\mathcal{Q}$ in a discrete manner. LUAVs select new patches to cover at the time step when the FUAVs finish covering the current patch. The patch selection problem is modeled as an episodic task with constant $T_0 > 0$ time steps in each episode. However, an episode can terminates early if the whole coverage task is completed within T_0 time steps.

States: The joint state of all LUAVs $\mathbf{s} \in \mathcal{S}$ is composed of three components, including the positions of all LUAVs, $\mathbf{s}_p$, the current coverage situation of the entire terrain $\mathbf{s}_c$, and the connectivity situation of LCN $\mathbf{s}_{con}$, i.e., $\mathbf{s} = (\mathbf{s}_p, \mathbf{s}_c, \mathbf{s}_{con})$. The position component $\mathbf{s}_p = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N]$ contains the positions of all LUAVs, where $\mathbf{p}_i = (x_i, y_i, z_i)$ is the 3D coordinates of LUAV L_i. Coverage situation component consists of the coverage matrix of all sub-swarms, i.e., $\mathbf{s}_c = [\mathbf{C}_1, \mathbf{C}_2, \dots \mathbf{C}_N]$. Moreover, $\mathbf{s}_{con}$ is the connectivity matrix of LCN $\mathbf{V}$. Note that the initial state of LUAVs $\mathbf{s}_0$ is fixed, i.e., $\mathbf{s}_0 = (\mathbf{s}_{p,0}, \mathbf{s}_{c,0}, \mathbf{s}_{con,0})$, where $\mathbf{s}_{p,0}$, $\mathbf{s}_{c,0}$ and $\mathbf{s}_{con,0}$ represent the initial positions of LUAVs, initial coverage matrices and the initial connectivity matrix of LUAVs.

Observations: Each LUAV can only observe the information of itself and its neighbors at each time step. The probability function of observations of LUAV L_i is a deterministic function of the index of LUAV L_i and joint state of all LUAVs, i.e.,

$$\mathcal{O}(\mathbf{o}|\mathbf{s}, i) = \begin{cases} 1, & \text{if } \mathbf{o} = \mathbf{o}_i; \\ 0, & \text{otherwise,} \end{cases} \tag{33}$$

where $\mathbf{o}_i \in \Omega_i$ is denoted by a tuple $(\mathbf{p}_i, \mathbf{C}_i, \mathbf{p}_i^{nei}, \mathbf{C}_i^{nei})$. Note that $\mathbf{p}_i$ is the 3D positions of LUAV L_i and $\mathbf{C}_i$ is the coverage matrix of L_i. The remaining elements $\mathbf{p}_i^{nei} = (\mathbf{p}_i^1, \mathbf{p}_i^2, \dots, \mathbf{p}_i^{n_i})$ and $\mathbf{C}_i^{nei} = (\mathbf{C}_i^1, \mathbf{C}_i^2, \dots, \mathbf{C}_i^{n_i})$ represent the 3D positions and coverage matrices of the neighbors of L_i, respectively.

Actions: The action space is the same for all LUAVs, i.e., $\mathcal{A}_i = \{a_f, a_b, a_r, a_l\}, \forall i \in \mathcal{N}$. Note that the elements a_f, a_b, a_r and a_l represent the actions of selecting the forward, backward, right and left patch, respectively. At each time step, LUAV L_i selects an action from $\mathcal{A}_i$.

Rewards: The reward functions play an important role in RL algorithms, since reward functions can guide the agents to achieve the goals of the task. At each time step t, the common reward function R_t of all LUAVs is composed of a shared reward part $R_{sh,t}$ and private reward parts of N individual LUAVs, $R_{i,t}^{pv}$, $i \in \mathcal{N}$. Specifically, the shared reward $R_{sh,t}$ assesses the overall coverage rate of the entire terrain and connectivity conditions of the LCN at current time step t, and can be expressed as

$$R_{sh,t} = \beta_s r_{a,t}^c - \beta_c f_{c,t} - \beta_t, \tag{34}$$

where r_a^c is the overall coverage rate at time step t defined in (15), $f_{c,t}$ equals to 0 if the LCN at time step t is a connected graph while $f_{c,t}$ equals to 1 otherwise; $\beta_c > 0$ represents the punishment factor for the disconnection of LCN; $\beta_t > 0$ represents the punishment factor for consuming one time step. The private reward $R_{i,t}^{pv}$ is designed to be the punishment of LUAV L_i for revisiting the patches covered previously at time step t and can be expressed as

$$R_i^{pv} = \begin{cases} -\beta_{pv}, & \text{if } c_{i,t} = 1; \\ 0, & \text{otherwise,} \end{cases} \tag{35}$$

where $c_{i,t} = 1$ denotes the patch selected by LUAV L_i has been covered already before time step t. Note that β_s, β_{pv} are both positive constants. The common reward R_t is calculated as

$$R_t = R_{sh,t} + \sum_{i=1}^{N} R_{i,t}^{pv}. \tag{36}$$

Transition Probability and γ: The transition probability function $\mathcal{P}$ represents the dynamic characteristic of the environment. We assume there is no external disturbance from the environment for UAVs during the flight, and the sub-swarms can always successfully fly to the target patch selected by LUAVs. The discounted factor γ is endowed to be 1, i.e., $\gamma = 1$.

Denote $\pi = (\pi_1, \pi_2, \dots, \pi_N)$ as the joint policy of LUAVs, where π_i is the local policy of LUAV L_i. The expected return of the patch selection problem, $J_\pi(\mathbf{s}_0)$, is defined as the expected summation of the discounted common rewards in

one episode following joint policy π, i.e.,

$$J_\pi(\mathbf{s}_0) = \mathbb{E}_\pi\left[\sum_{t=0}^{T_0}\gamma^t R_t\right]. \tag{37}$$

Substitute the expression of reward function R_t and discounted factor γ into (37), the expected return can be expressed as

$$\begin{aligned} J_\pi(\mathbf{s}_0) &= \mathbb{E}_\pi\left[\sum_{t=0}^{T_0}\gamma^t\left[R_{sh,t}+\sum_{i=1}^{N}R_{i,t}^{pv}\right]\right]_{\gamma=1} \\ &= \mathbb{E}_\pi\left[\sum_{t=0}^{T}\gamma^t\left[\beta_s r_{a,t}^c-\beta_c f_{c,t}-\beta_t+\sum_{i=1}^{N}R_{i,t}^{pv}\right]\right]_{\gamma=1} \\ &= -\beta_t T+\mathbb{E}_\pi\left[\beta_s\sum_{t=0}^{T}r_{a,t}^c\right]+\mathbb{E}_\pi\Bigg[-\beta_c\sum_{t=0}^{T}f_{c,t} \\ &\quad +\sum_{t=0}^{T}\sum_{i=1}^{N}R_{i,t}^{pv}\Bigg]. \end{aligned} \tag{38}$$

As the LUAV patch selection problem is a cooperative problem, the goal of Dec-POMDP is designed to find an optimal joint policy $\pi^* = (\pi_1^*, \pi_2^*, \ldots, \pi_N^*)$ that maximizes the expected return $J_\pi(\mathbf{s}_0)$, i.e.,

$$\pi^* = \arg\max_\pi J_\pi(\mathbf{s}_0), \tag{39}$$

Note that term $\mathbb{E}_\pi\Big[-\beta_c\sum_{t=0}^{T}f_{c,t}+\sum_{t=0}^{T}\sum_{i=1}^{N}R_{i,t}^{pv}\Big]$ in $J_\pi(\mathbf{s}_0)$ is the punishment term for disconnections of LCN and revisiting of covered patches, and is equivalent to the constraints in (18a) and (18b). Under optimal joint policy π^*, the punishment term in $J_\pi(\mathbf{s}_0)$ equals to zero, i.e.,

$$\mathbb{E}_{\pi^*}\left[-\beta_c\sum_{t=0}^{T}f_{c,t}+\sum_{t=0}^{T}\sum_{i=1}^{N}R_{i,t}^{pv}\right]=0, \tag{40}$$

and the optimal expected return can be expressed as

$$J_{\pi^*}(\mathbf{s}_0) = -\beta_t T_* + \mathbb{E}_{\pi^*}\left[\beta_s\sum_{t=0}^{T}r_{a,t}^c\right]. \tag{41}$$

In fact, the expected return $J_\pi(\mathbf{s}_0)$ has the same form as the Lagrange equation of (18). Hence, maximizing the expected return J_π in Dec-POMDP is consistent with achieving the goals of the LUAV patch selection problem, i.e., covering as much area as possible in the shortest possible time, and the modelled Dec-POMDP problem is equivalent to the LUAV patch selection problem.

Note that although we are searching for global optimal policy π^*, the resulted policy can probability only be a *Nash Equilibrium* [35], [36] of Dec-POMDP due to the *value decomposition structure* we used in the following algorithm [37].

B. Observation History Design

Denote $\tau_t = (\tau_{1,t}, \tau_{2,t}, \ldots, \tau_{N,t})$ as the joint observation history of all LUAVs, where $\tau_{i,t} \in \mathcal{T}_i = (\Omega_i, \mathcal{A}_i)^*$ is the observation history of LUAV L_i at time step t. Each observation history $\tau_{i,t}$ needs to contain the history information of observations and actions of LUAV L_i. In multi-agent DRL, the observation history is formed by directly concatenating the observations in past consecutive $(u+1)$ time steps, i.e.,

$$\tau_{i,t} = (o_{i,(t-u)}, o_{i,(t-u+1)}, \ldots, o_{i,t}), \tag{42}$$

where $0 < u \leq t$ represents a previous time point, and $o_{i,t}$ is the observation of LUAV L_i at time step t. There are three main issues when directly concatenating the historical observations to form $\tau_{i,t}$:

1) **High dimensionality:** Different from multi-agent RL, the number of agents can be extremely large in a UAV swarm. Hence, the dimension of observation history can be very high, which can cause difficulties in policy exploration during training process.
2) **Dynamic changing dimensions:** As the neighbors of each LUAV change dynamically during the task, the number of neighbors for one LUAV at different time steps may not be the same. Therefore, the dimension of the observation history changes over time, which will increase the difficulties of designing the policy model of LUAVs.
3) **Various observation dimensions:** Since the dimensions of elements in observation $\mathbf{o}_i = (\mathbf{p}_i, \mathbf{C}_i, \mathbf{p}_i^{nei}, \mathbf{C}_i^{nei})$ are different, directly concatenating of these elements can lose information.

To solve the above problems, we design the structure of observation history $\tau_{i,t}$ for each LUAV L_i based on the characteristics of the terrain model and communication features of the LCN, as shown in Fig. 6. The observation history of each LUAV L_i is the concatenation of three vectors, i.e.,

$$\tau_{i,t} = (\tau_{i,t}^p, \tau_{i,t}^{nei,p}, \tau_{i,t}^c), \tag{43}$$

where $\tau_{i,t}^p$ and $\tau_{i,t}^{nei,p}$ represent the history position information of LUAV L_i and the neighbors of LUAV L_i at time step t, respectively, and $\tau_{i,t}^c$ represents the coverage history of LUAV L_i at time step t.

1) Design of $\tau_{i,t}^p$ and $\tau_{i,t}^{nei,p}$: The blocks with width 1 and length 2 in Fig. 6 represent the history positions of LUAV L_i and the neighbors of LUAV L_i, and each block is a 3D coordinate vector. The two balls with "+" symbols next to the position blocks both represent the mean operators. The current $\tau_{i,t}^p$ of LUAV L_i is calculated by taking the average of the positions of LUAV L_i in the past $(u+1)$ time steps, i.e.,

$$\tau_{i,t}^p = \frac{1}{u+1}\sum_{t'=t-u}^{t}\left[\mathbf{p}_i^{t'}\right]. \tag{44}$$

For the positions of the neighbors of LUAV L_i, we use the mean embedding method [38] to extract their history position information as

$$\tau_{i,t}^{nei,p} = \mathbb{E}_{t'}\left[\phi(\mathbf{p}_{i,t'}^{nei})\right], \tag{45}$$

where ϕ is a feature mapping, and $t' \in \{t-u, t-u+1, \ldots, t\}$ is a uniformly distributed random variable. We denote ϕ as the average value of all n_i neighbors of LUAV L_i, and $\tau_{i,t}^{nei,p}$ can be expressed as

$$\tau_{i,t}^{nei,p} = \frac{1}{u+1}\sum_{t'=t-u}^{t}\frac{1}{n_i}\sum_{j=1}^{n_i}\left[\mathbf{p}_{i,t'}^{nei,j}\right]. \tag{46}$$

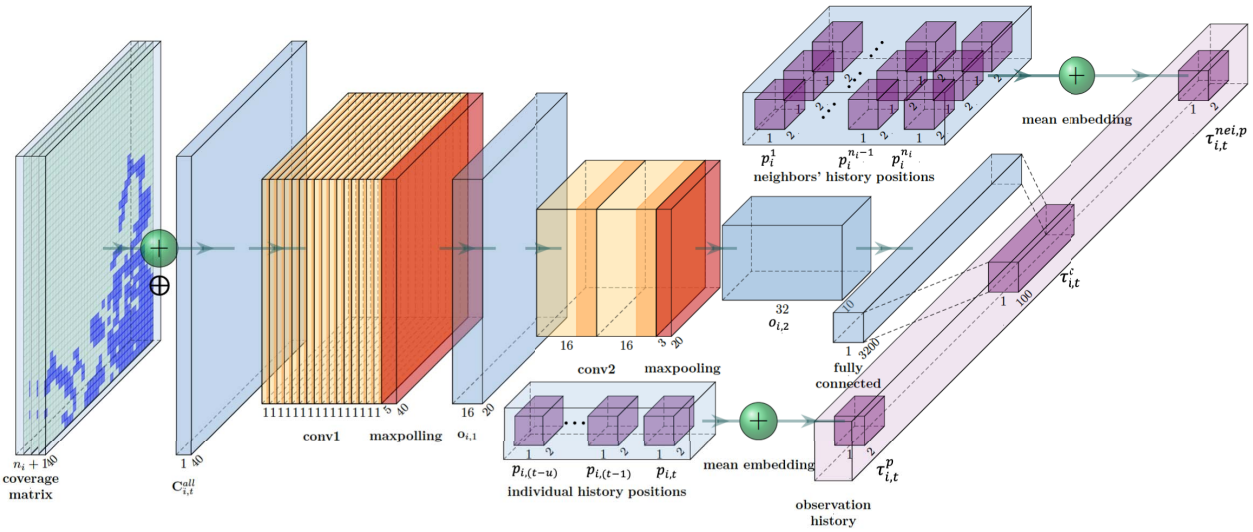


Fig. 6. Observation history model for each LUAV L_i. The observation history is composed of three parts, including the coverage information of LUAV and its neighbors $\tau^c_{i,t}$ processed by CNNs, the mean embedding of history positions of LUAV itself, $\tau^p_{i,t}$, and the mean embedding of the neighbor LUAVs' history positions, $\tau^{nei,p}_{i,t}$.

Note that the dimensions of both $\tau^{nei,p}_i$ and τ^p_i remain constant values, even though the number of neighbors n_i changes during the task.

2) Design of $\tau^c_{i,t}$: The coverage information includes the history coverage matrices of LUAV L_i and its neighbors. Here we only use the coverage matrices of current time step t, $(\mathbf{C}_{i,t}, \mathbf{C}^1_{i,t}, \ldots, \mathbf{C}^{n_i}_{i,t})$, as they have already contained all the covered patches information up to now. Note that $\mathbf{C}_{i,t}$ represents the coverage matrix of L_i at time step t, and $\mathbf{C}^j_{i,t}$ represents the coverage matrix of the j-th neighbor of L_i at time step t. We first define a *matrix or* operator.

Definition 2 (Matrix or): Given two $N_x \times N_y$ coverage matrices $\mathbf{C}' = (c'_{ij})$ and $\mathbf{C}'' = (c''_{ij})$, the matrix or operation of $\mathbf{C}'$ and $\mathbf{C}''$ is also an $N_x \times N_y$ coverage matrix $\mathbf{C} = (c_{ij})$, where $c_{ij} = c'_{ij} \circ c''_{ij}$ is the logical or between c'_{ij} and c''_{ij}. Denote the matrix or operator as $\oplus$, i.e., $\mathbf{C} = \mathbf{C}' \oplus \mathbf{C}''$.

Firstly, we take the matrix or operation of all the current coverage matrices, i.e.,

$$\mathbf{C}^{all}_{i,t} = \mathbf{C}_{i,t} \oplus \mathbf{C}^1_{i,t} \ldots \oplus \mathbf{C}^{n_i}_{i,t}, \tag{47}$$

where $\mathbf{C}^{all}_{i,t}$ is the new coverage matrix that gathers all the covered patches information of LUAV L_i and its neighbors. Note that no matter how large the number of LUAV L_i's neighbors, n_i, is, and no matter how n_i changes during the flight, $\mathbf{C}^{all}_{i,t}$ remains an $N_x \times N_y$ coverage matrix. The invariant dimensions of $\mathbf{C}^{all}_{i,t}$ solve the *large number of agents* issue and *dynamic changing neighbors* issue described above.

Secondly, as $\mathbf{C}^{all}_{i,t}$ is a 2D matrix that is difficult to be concatenated with the position observation history vectors $\tau^p_{i,t}$ and $\tau^{nei,p}_{i,t}$, we use a shallow CNN to extract the features of $\mathbf{C}^{all}_{i,t}$. Specifically, as shown in Fig. 6, the CNN contains two kinds of layers, including convoluted layers and fully connected layers that are responsible for turning the 2D matrix into a vector. Detailed architecture design of the CNN depends on the size of inputs $\mathbf{C}^{all}_{i,t}$, and will be further described in the simulations (Section VI-B). The output of CNN is $\tau^c_{i,t}$ that is the third element of the designed observation history $\tau_{i,t}$, i.e., $\tau^c_{i,t} = \text{CNN}(\mathbf{C}^{all}_{i,t})$, where $\text{CNN}(\cdot)$ represents the convolution neural network operation. Note that $\tau^c_{i,t}$ has the same dimension with $\tau^{nei,p}_{i,t}$ and $\tau^p_{i,t}$, which solve s the *various observation dimensions* issue.

C. Swarm Deep Q-Learning

To find the optimal policy π^* of Dec-POMDP, we propose a swarm deep Q-learning (SDQN) algorithm with the *centralized training decentralized execution* (CTDE) framework. Similar to single-agent reinforcement learning algorithms, we construct a global value function $Q_\pi(\tau, \mathbf{a})$ for LUAVs under certain policy π as

$$Q_\pi(\tau, \mathbf{a}) = \mathbb{E}_\pi\left[\sum_{t=t_0}^{T} \gamma^{t-t_0} R_t | \tau_{t_0} = \tau, \mathbf{a}_{t_0} = \mathbf{a}\right], \tag{48}$$

where t_0 represents the time step when the global observation history is τ and joint action is $\mathbf{a}$. We assign each LUAV L_i a local value function $Q_{\pi,i}(\tau, \mathbf{a})$, and the global value function $Q_\pi(\tau, \mathbf{a})$ can be expressed as

$$Q_\pi(\tau, \mathbf{a}) = f(Q_{\pi,1}(\tau, \mathbf{a}), \ldots, Q_{\pi,N}(\tau, \mathbf{a})), \tag{49}$$

where $f : \mathbb{R}^N \to \mathbb{R}$ is a valid factorization function. However, since τ and $\mathbf{a}$ cannot be obtained by each LUAV L_i at each time step, the variables of local value function $Q_{\pi,i}$ can only be the local observation history τ_i and local action $\mathbf{a}_i$. We assume that $Q_{\pi,i}(\tau_i, \mathbf{a}_i)$ can approximate the local value function $Q_{\pi,i}(\tau, \mathbf{a})$ for LUAV L_i, i.e. , [37],

$$Q_{\pi,i}(\tau, \mathbf{a}) \approx Q_{\pi,i}(\tau_i, \mathbf{a}_i). \tag{50}$$

This assumption is consistent with the considered system models to a great extent. Recall that we divide lanes in terrain $\mathcal{Q}$ for each LUAV in advance. LUAVs can only cover the patches within its own lane before completing the coverage of the lane and ignore the coverage behaviors of its neighbors.

Therefore, each LUAV L_i can make decisions only based on τ_i and $\mathbf{a}_i$ during the period of covering its designed lane, which indicates that the local value function $Q_{\pi,i}$ depends more on the information of itself in the task. Hence, we can safely say that

$$Q_{\pi,i}(\tau, \mathbf{a}) = Q_{\pi,i}(\tau_i, \mathbf{a}_i). \tag{51}$$

In the later period of the task when LUAVs cooperate with others, the local value function of each LUAV L_i depends more on the information of LUAV L_i and the neighbors of LUAV L_i, and can be expressed as

$$Q_{\pi,i}(\tau, \mathbf{a}) = Q_{\pi,i}(\tau_i, \mathbf{a}_i, \tau_i^{nei,1}, \mathbf{a}_i^{nei,2}, \ldots, \tau_i^{nei,n_i}, \mathbf{a}_i^{nei,n_i}), \tag{52}$$

where $\tau_i^{nei,j}$ represents the observation history of the j-th neighbor of LUAV L_i. However, as the observation history τ_i already contains the position and coverage information of LUAV L_i and its neighbors, the observation histories of neighbor LUAVs, i.e., $\tau_i^{nei,j},\ j \in \mathcal{N}(i)$, cannot provide more information than τ_i. Therefore, the local value function of LUAV L_i can be expressed as

$$\begin{aligned} &Q_{\pi,i}(\tau_i, \mathbf{a}_i, \tau_i^{nei,1}, \mathbf{a}_i^{nei,2}, \ldots, \tau_i^{nei,n_i}, \mathbf{a}_i^{nei,n_i}) \\ &\quad = Q_{\pi,i}(\tau_i, \mathbf{a}_i, \mathbf{a}_i^{nei,2}, \ldots, \mathbf{a}_i^{nei,n_i}) \approx Q_{\pi,i}(\tau_i, \mathbf{a}_i), \end{aligned} \tag{53}$$

where the approximation is based on the assumption that LUAVs focus more on the actions of themselves than the actions of their neighbors. Hence, (50) is satisfied during the overall period of the task. We replace $Q_{\pi,i}(\tau, \mathbf{a})$ by $Q_{\pi,i}(\tau_i, \mathbf{a}_i)$ as the local value function of LUAV L_i in the following context.

To design a valid factorization function f, the local value functions should satisfy the *individual global max* (IGM) principle [39], i.e.,

$$\arg\max_{\mathbf{a}} Q_\pi(\tau, \mathbf{a}) = \begin{pmatrix} \arg\max_{\mathbf{a}_1} Q_{\pi,1}(\tau_1, \mathbf{a}_1) \\ \cdots \\ \arg\max_{\mathbf{a_N}} Q_{\pi,N}(\tau_N, \mathbf{a}_N) \end{pmatrix}. \tag{54}$$

IGM indicates that the combination of the local optimal actions is the optimal global joint action. In the proposed SDQN algorithm, we use a simple summation function [37] as the valid factorization function, i.e.,

$$Q_\pi(\tau, \mathbf{a}) = \sum_{i=1}^{N} Q_{\pi,i}(\tau_i, \mathbf{a}_i). \tag{55}$$

Note that the summation factorization function is satisfied with (56) and (57), as shown at the bottom of the page, the IGM principle, since

$$\begin{aligned} \sum_{i=1}^{N} \max_{\mathbf{a}_i} Q_{\pi,i}(\tau_i, \mathbf{a}_i) &= \max_{\mathbf{a}} \sum_{i=1}^{N} Q_{\pi,i}(\tau_i, \mathbf{a}_i) \\ &= \max_{\mathbf{a}} Q_\pi(\tau, \mathbf{a}). \end{aligned} \tag{58}$$

We use a neural network (NN), $Q_{\pi,i}(\tau_i, \mathbf{a}_i; \alpha_i)$, that composed of several fully connected layers to approximate the local value function for each LUAV L_i, where α_i is the training parameter. Note that the individual policy can be derived directly from $Q_{\pi,i}(\tau_i, \mathbf{a}_i; \alpha_i)$, i.e.,

$$\mathbf{a}_i = \pi_i(\tau_i) = \max_{\mathbf{a}_i \in \mathcal{A}_i} Q_{\pi,i}(\tau_i, \mathbf{a}_i; \alpha_i). \tag{59}$$

The overall structure of SDQN algorithm is shown in Fig. 7, where τ_i is the output of the observation history model of LUAV L_i in Fig. 6.

Inspired by DQN [40], we set a replay buffer $\mathcal{D}$ with capacity $|\mathcal{D}|$ to store the experience of LUAVs in training local value functions $Q_{\pi,i}(\tau_i, \mathbf{a}_i; \alpha_i),\ i \in \mathcal{N}$. Specifically, the k-th experience s $\tilde{\mathbf{e}}^k \in \mathcal{D}$ is a tuple,

$$\begin{aligned} \mathcal{L}(\alpha_i) &= \mathbb{E}_{\tilde{\mathbf{e}} \sim \mathcal{D}}\left[\left(\tilde{R} + \gamma \max_{\tilde{\mathbf{a}}'} \sum_{i=1}^{N} Q_{\pi,i}^{target}(\tilde{\tau}_i', \tilde{\mathbf{a}}_i'; \beta_i) - \sum_{i=1}^{N} Q_{\pi,i}(\tilde{\tau}_i, \tilde{\mathbf{a}}_i; \alpha_i)\right)^2\right] \\ &= \mathbb{E}_{\tilde{\mathbf{e}} \sim \mathcal{D}}\left[\left(\tilde{R} + \gamma \sum_{i=1}^{N} \max_{\tilde{\mathbf{a}}_i'} Q_{\pi,i}^{target}(\tilde{\tau}_i', \tilde{\mathbf{a}}_i'; \beta_i) - \sum_{i=1}^{N} Q_{\pi,i}(\tilde{\tau}_i, \tilde{\mathbf{a}}_i; \alpha_i)\right)^2\right]. \end{aligned} \tag{56}$$

$$\begin{aligned} \nabla_{\alpha_i}\mathcal{L}(\alpha_i) &= \nabla_{\alpha_i}\mathbb{E}_{\tilde{\mathbf{e}}_i \sim \mathcal{D}}\left[\left(\tilde{R} + \gamma \sum_{i=1}^{N} \max_{\tilde{\mathbf{a}}_i'} Q_{\pi,i}^{target}(\tilde{\tau}_i', \tilde{\mathbf{a}}_i'; \beta_i) - \sum_{i=1}^{N} Q_{\pi,i}(\tilde{\tau}_i, \tilde{\mathbf{a}}_i; \alpha_i)\right)^2\right] \\ &= \mathbb{E}_{\tilde{\mathbf{e}}_i \sim \mathcal{D}}\left[\nabla_{\alpha_i}\left(\tilde{R} + \gamma \sum_{i=1}^{N} \max_{\tilde{\mathbf{a}}_i'} Q_{\pi,i}^{target}(\tilde{\tau}_i', \tilde{\mathbf{a}}_i'; \beta_i) - \sum_{i=1}^{N} Q_{\pi,i}(\tilde{\tau}_i, \tilde{\mathbf{a}}_i; \alpha_i)\right)^2\right] \\ &= \mathbb{E}_{\tilde{\mathbf{e}}_i \sim \mathcal{D}}\left[2\left(\tilde{R} + \gamma \sum_{i=1}^{N} \max_{\tilde{\mathbf{a}}_i'} Q_{\pi,i}^{target}(\tilde{\tau}_i', \tilde{\mathbf{a}}_i'; \beta_i) - \sum_{i=1}^{N} Q_{\pi,i}(\tilde{\tau}_i, \tilde{\mathbf{a}}_i; \alpha_i)\right)\right. \\ &\quad \left. \times \nabla_{\alpha_i}\left(\tilde{R} + \gamma \sum_{i=1}^{N} \max_{\tilde{\mathbf{a}}_i'} Q_{\pi,i}^{target}(\tilde{\tau}_i', \tilde{\mathbf{a}}_i'; \beta_i) - \sum_{i=1}^{N} Q_{\pi,i}(\tilde{\tau}_i, \tilde{\mathbf{a}}_i; \alpha_i)\right)\right] \\ &= \mathbb{E}_{\tilde{\mathbf{e}}_i \sim \mathcal{D}}\left[-2\left(\tilde{R} + \gamma \sum_{i=1}^{N} \max_{\tilde{\mathbf{a}}_i'} Q_{\pi,i}^{target}(\tilde{\tau}_i', \tilde{\mathbf{a}}_i'; \beta_i) - \sum_{i=1}^{N} Q_{\pi,i}(\tilde{\tau}_i, \tilde{\mathbf{a}}_i; \alpha_i)\right) \times \nabla_{\alpha_i} Q_{\pi,i}(\tilde{\tau}_i, \tilde{\mathbf{a}}_i; \alpha_i)\right]. \end{aligned} \tag{57}$$

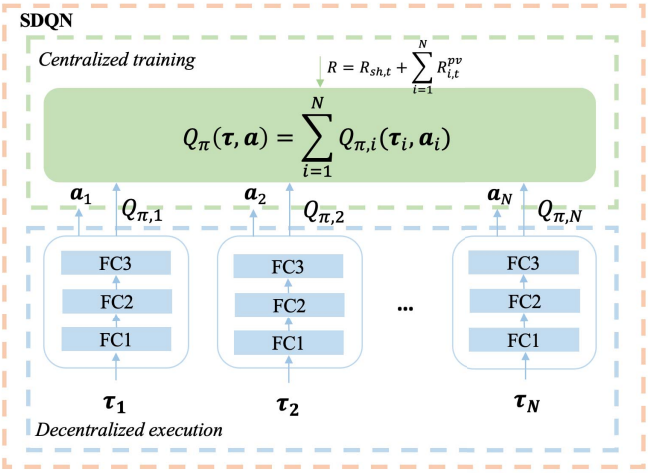


Fig. 7. SDQN algorithm based on value decomposition method using centralized training and decentralized execution framework.

i.e., $\tilde{\mathbf{e}}^k = (\tilde{\tau}_1^k, \tilde{\mathbf{a}}_1^k, \tilde{\tau}_1^{'k}, \tilde{\tau}_2^k, \tilde{\mathbf{a}}_2^k, \tilde{\tau}_2^{'k}, \ldots, \tilde{\tau}_N^k, \tilde{\mathbf{a}}_N^k, \tilde{\tau}_N^{'k}, \tilde{R}^k)$, $k \in \{1, 2, \ldots, |\mathcal{D}|\}$, where $\tilde{\tau}_i^k$, $\tilde{\mathbf{a}}_i^k$, and $\tilde{\tau}_i^{'k}$ represent the observation history, action and next observation history of LUAV L_i, while $\tilde{R}^k$ represents the common reward of all N agents. Based on (58), the loss function of the i-th local value function $Q_{\pi,i}(\tau_i, \mathbf{a}_i; \alpha_i)$ can be expressed as (56) on the bottom of the previous page, where $Q_{\pi,i}^{target}$ is the target local history value function of LUAV L_i with parameter β_i. The gradient of the loss function $\nabla_{\alpha_i}\mathcal{L}_i(\alpha_i)$ is calculated as (57) on the bottom of the previous page, and the update rule of α_i is

$$\alpha_i^{new} \leftarrow \alpha_i^{old} - \kappa_i \nabla_{\alpha_i}\mathcal{L}(\alpha_i), \tag{60}$$

where α_i^{old} is the original parameter, α_i^{new} is the updated parameter, and $\kappa_i > 0$ is the learning rate. During the training stage, to break the correlations between different experiences, we randomly sample a batch of $|\mathcal{B}|$ experiences from the $\mathcal{D}$. Hence, the gradient of the loss function can be expressed as

$$\nabla_{\alpha_i} L(\alpha_i) = -\frac{2}{|\mathcal{B}|}\sum_{k=1}^{|\mathcal{B}|}\left(\tilde{R}_i^k + \gamma \sum_{i=1}^{N} \max_{\tilde{\mathbf{a}}_i^{'k}} Q_{\pi,i}^{target}(\tilde{\tau}_i^{'k}, \tilde{\mathbf{a}}_i^{'k}; \beta_i) - \sum_{i=1}^{N} Q_{\pi,i}(\tilde{\tau}_i^k, \tilde{\mathbf{a}}_i^k; \alpha_i)\right) \times \nabla_{\alpha_i} Q_{\pi,i}(\tilde{\tau}_i^k, \tilde{\mathbf{a}}_i^k; \alpha_i). \tag{61}$$

Note that the target value function $Q_{\pi,i}^{target}(\tilde{\tau}_i, \tilde{\mathbf{a}}_i; \beta_i)$ is updated every episode using the soft update rule, i.e.,

$$\beta_i^{new} \leftarrow \theta\beta_i^{old} + (1-\theta)\alpha_i, \tag{62}$$

where $\theta \in (0, 1)$ is the constant update rate. The soft update rule can increase the robustness of training.

During the patch selection process, LUAVs communicate with their neighbors constantly, and the communication overloads for LUAV L_i can be expressed as $\mathcal{O}(\sum_{t=0}^{T}|\mathcal{N}_t(i)|) \leq \mathcal{O}(TN)$. The total communication overloads for each LUAV is $\mathcal{O}(T(M+N))$. Note that there is no communication overload of FUAVs during the patch selection process. Hence, the maximum communication overload among all UAVs is smaller than the scale of UAV swarm, i.e.,

$$\max\{\mathcal{O}(T(M+N)), \mathcal{O}(TM)\} < \mathcal{O}(TN(M+1)). \tag{63}$$

The algorithms we proposed are communication efficient.

VI. Simulation Results

The terrain $\mathcal{Q}$ in the simulation is shown in Fig. 2, which is a 1,000m×1,000m rectangular shaped mountain with an irregular surface. The width and length of rectangles in regular grid are both set to be 25m, i.e., $\Delta x = \Delta y = 25$m. Hence, the number of rectangles is

$$N_x \times N_y = \left\lceil \frac{L_x}{\Delta x} \right\rceil \times \left\lceil \frac{L_y}{\Delta y} \right\rceil = 40 \times 40 = 1,600. \tag{64}$$

The scattered points on the terrain are the sampling points of the regular grid method, and the number of sampling points is $41 \times 41 = 1,681$. The terrain $\mathcal{Q}$ is divided into 1,600 patches. The areas of patches are calculated by the proposed geometric method in Section III-A. The color depth d_{kl} in Fig. 3 is in the range of $[1, 4)$, $\forall k, l \in [1, 40]$. There are 10 sub-swarms in the UAV swarm while each sub-swarm contains one LUAV and four FUAVs. The total number of UAVs is $(4+1) \times 10 = 50$. The starting positions of sub-swarms are all along X axis, separated by $40/10 = 4$ patches (or equivalently 100m). The lengths of the panels for sub-swarms are all set to be 750m. The communication range of LUAVs is 250m [41], i.e., $R_{cl} = 250$m. All the UAVs are always above the surface of $\mathcal{Q}$ at a fixed height $H = 5$m. The coverage radius of all FUAVs are set to be 1m, and the flying speed of each FUAV is 1m/s. Besides, the threshold of replicated coverage rate θ_c is set to be 5%. We implement CNN and NNs of SDQN based on PyTorch framework. The training process is conducted on CPU, Intel(R) Core(TM) i7-10700K @3.80GHz (16 CPUs), with 32GB memory, accelerated by GPU, NVIDIA GeForce RTX 2080 Super. One time step in the simulation corresponds to one second in the real world.

A. Patch Coverage Trajectories of FUAVs

In the simulations, the FUAVs are responsible for carrying out the coverage of the patches selected by their LUAVs. The trajectories of FUAVs within a patch are designed according to the proposed trajectory planning algorithm in Section III-B. We take a patch $ABCD$ with width 29m and length 30m (total area 870 m^2) as an example. Fig. 8 shows the detailed coverage trajectories of the four FUAVs within a sub-swarm, where the colored circles represent the coverage circles with radius 1m of FUAVs at each time step, and the arrows represent the moving directions of the FUAVs. The starting positions of four FUAVs are at the diamonds locations. The proposed algorithm divides the patch into 4 cells with equal areas ($217.5m^2$). Each FUAV covers its own cell using spiral-Zigzag searching pattern. The flying distances of the FUAVs 1 and 3 are both 120.07m while the distances of FUAVs 2 and 4 are both 125.39m. The total coverage time of patch $ABCD$ is 125.39 time steps. The FUAVs cover all the areas of patch $ABCD$. Each FUAV has a small redundant coverage area of

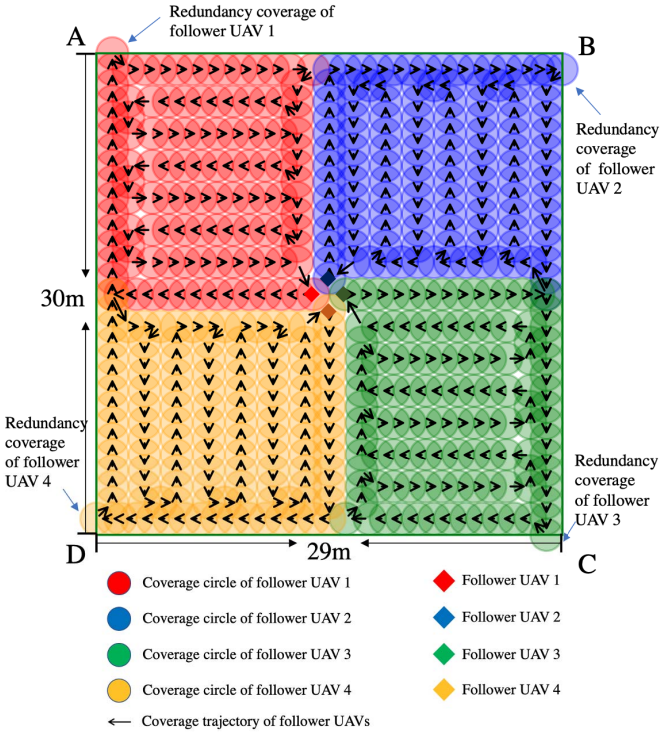


Fig. 8. Four FUAVs' coverage trajectories of a single patch $ABCD$. The circles represent the coverage ranges of FUAVs at different time steps, and the arrows represent the moving directions of FUAVs.

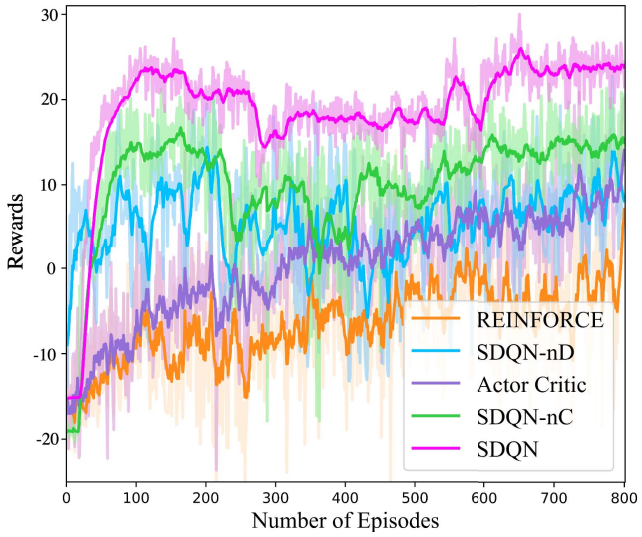


Fig. 9. Training returns of episodes (or equivalently $J_\pi(\mathbf{s}_0)$) of different algorithms. SDQN outperforms traditional reinforcement learning algorithms, including REINFORCE and actor critic.

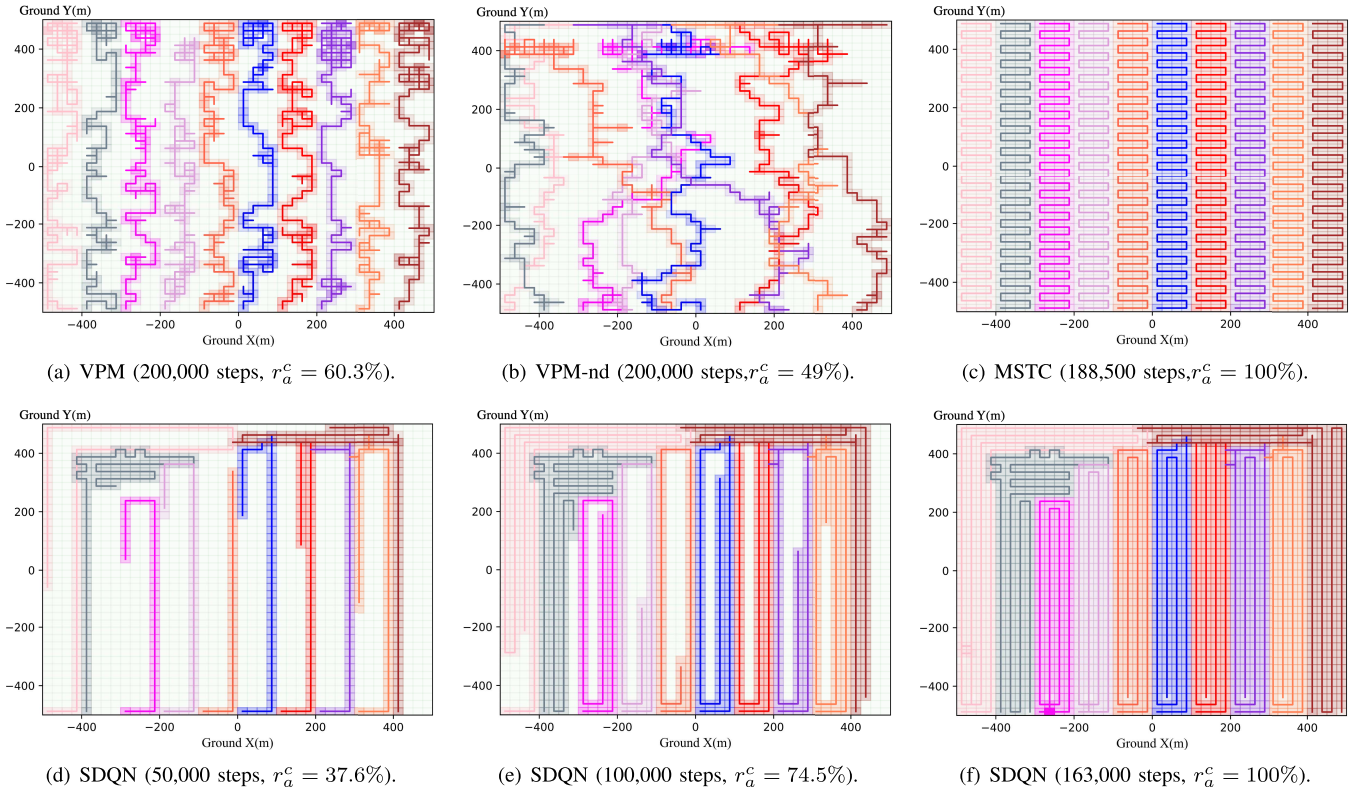
$\frac{\pi}{2}m^2$, and the total redundant coverage area of four FUAVs is only 0.721% of the area of patch $ABCD$. The results indicate that the proposed trajectory planning algorithm of FUAVs is feasible and efficient.

B. Patch Selections of LUAVs

LUAVs are responsible for selecting patches for FUAVs to cover. We mainly apply the proposed SDQN algorithm to LUAVs. In the observation history model of SDQN, the replay time u is set to be 2s. The input of CNN can be viewed as a 40×40 image with one channel. The structures of CNN contain three layers, including two convolutional layers and one fully connected layer, as shown in Fig. 6. The first layer is a convolutional layer with sixteen 5×5 kernels (or feature maps) followed by a maxpooling layer and ReLU activation functions. The output of the first layer is a 20×20 image with 16 channels. The second layer is also a convolutional layer with two 3×3 kernels followed by a maxpooling layer and ReLU activation functions. The output of the second layer is a 10×10 image with 32 channels. The third layer is a fully connected layer that receives the output of the second layer and outputs a 1×100 vector. Since the positions of each LUAV and its neighbors are all 1×2 vectors, the whole observation history is a 1×104 vector. The action observation history value function $Q_{\pi,i}$ of each LUAV L_i is a fully connected NN with two hidden layers. The number of neurons of two hidden layers is 100 and 10, respectively, and the activation functions of both layers are ReLU functions. The output layer is a fully connected layer with log-softmax activation functions.

Fig. 9 shows the rewards of SDQN, the variants of SDQN and other RL algorithms during the training process. The number of training episodes is set to be 800 with 200,000 steps each. Note that SDQN-nC represents the SDQN algorithm with no CNN in observation history model, and SDQN-nD is the SDQN algorithm with no panel divisions of terrain $\mathcal{Q}$ in advance. From Fig. 9, we can see that the rewards of SDQN rise much more quickly than that of the other four algorithms. The final rewards of SDQN-nC are less than that of SDQN, which indicates that the CNN in observation history model correctly extracts the features of coverage information of each LUAV and its neighbors. Moreover, the rewards of SDQN-nD rise slower than that of both SDQN and SDQN-nC, which indicates that the panel divisions based on prior knowledge play an important part in the performance improvement. From the high vibrating rewards curve of SDQN-nD, we can see that the panel divisions will reduce the performance variance of LUAVs by increasing the disciplines of patch selections for LUAVs. Furthermore, SDQN has better performance than both Actor Critic and REINFORCE algorithms. The rewards of Actor Critic have lower variance than the rewards of REINFORCE, because Actor Critic algorithm uses an extra critic network to guide the improvement directions of policies.

In Fig. 10, we compare the proposed algorithms with two traditional multi-agent coverage algorithms, including vibrating particles model (VPM) [42] and multi-agent spanning tree coverage (MSTC) [43]. VPM is a heuristic algorithm inspired from bird flock foraging. A personal range p_r and a flock range $f_r \geq p_r$ are set for each agent in VPM algorithm. The basic idea of VPM is to make each agent get close to the agents within its flock range and avoid the agents within its personal range. The other algorithm MSTC is based on single-agent coverage algorithm spanning tree coverage. The basic idea of MSTC is to cover the areas along the edges of boundaries and obstacles.



(a) VPM (200,000 steps, $r_a^c = 60.3\%$). (b) VPM-nd (200,000 steps,$r_a^c = 49\%$). (c) MSTC (188,500 steps,$r_a^c = 100\%$).

(d) SDQN (50,000 steps, $r_a^c = 37.6\%$). (e) SDQN (100,000 steps, $r_a^c = 74.5\%$). (f) SDQN (163,000 steps, $r_a^c = 100\%$).

Fig. 10. Area coverage of different algorithms, where different colored patches and lines represent the covered patches and coverage trajectories of 10 different sub-swarms, respectively.

From Fig. 10(a) to Fig. 10(f), the 2D maps are the regular grids for constructing the 3D DEM of terrain, where each rectangle represents the corresponding patch of the terrain. Ten different colors represent the selections of ten sub-swarms. Specifically, the colored rectangles represent that the corresponding patches have been covered by the sub-swarm, while uncolored rectangles represent that the corresponding patches have not been selected or covered by any sub-swarm. Note that VPM-nd represents the VPM algorithm with no panel divisions. Fig. 10(a) and Fig. 10(b) represent the patch selections using VPM and VPM-nd, respectively. We can see that LUAVs repeatedly select the covered patches, and do not select all the patches by 200,000 time steps. Fig. 10(c) represents the patch selections of MSTC algorithm (with panel divisions in advance), and Fig. 10(d) to Fig. 10(f) show the patch selection process using SDQN algorithm. Both MSTC and SDQN algorithm make LUAVs successfully select all the patches of terrain. However, LUAVs using MSTC algorithm only select the patches of their own lanes without cooperation, which causes the coverage time of MSTC being longer than that of SDQN algorithm.

From Fig. 11, we can see that the overall coverage rates of the different algorithms change with times. We also see that SDQN is the first to complete full coverage of the terrain within 163,000 time steps, while MSTC algorithm completes the overall coverage by 188,500 time steps. Other algorithms, including SDQN-nC, SDQN-nD, VPM-nD and VPM, fail to make UAV swarm cover all the terrain. Note that the coverage rate curve of VPM in Fig. 11 is always above the curve of VPM-nD, which indicates that the proposed panels division

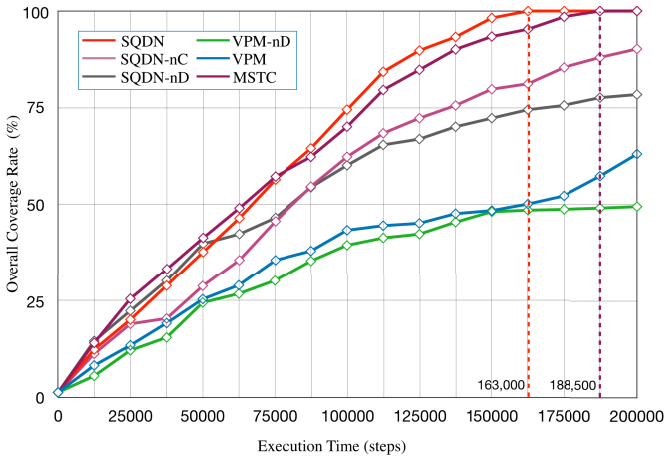


Fig. 11. Coverage rates of different algorithms over time. SDQN completes covering the terrain within 163,000 time steps, outperforming other traditional methods.

method is effective not only for SDQN algorithm but also for other traditional coverage algorithms.

C. Overall Coverage of UAV Swarm Using SDQN

The coverage rates of different sub-swarms using SDQN algorithm are shown in Fig. 12, where the valid individual coverage rates represent the valid covered areas without duplicate coverage. Note that the summation of all valid area coverage percentages equals to 100%, since the UAV swarm completes covering the entire terrain. We can see that each sub-swarm nearly covers 10% area of the entire terrain, which

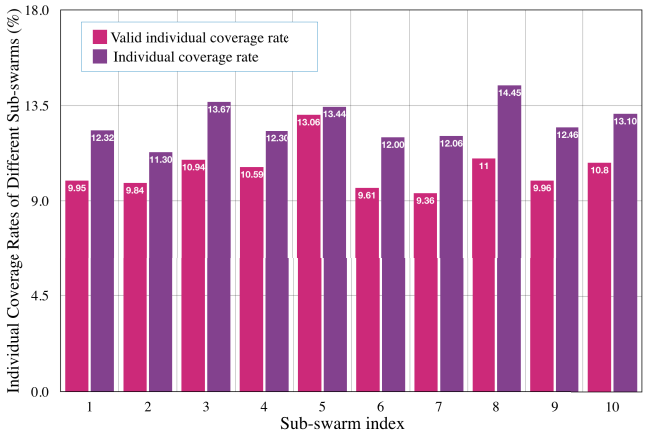


Fig. 12. Coverage rates of different sub-swarms using SDQN. The repeated coverage rate $r_p^c = 2.71\% < 5\%$ indicates low repeated coverage of UAV swarm.

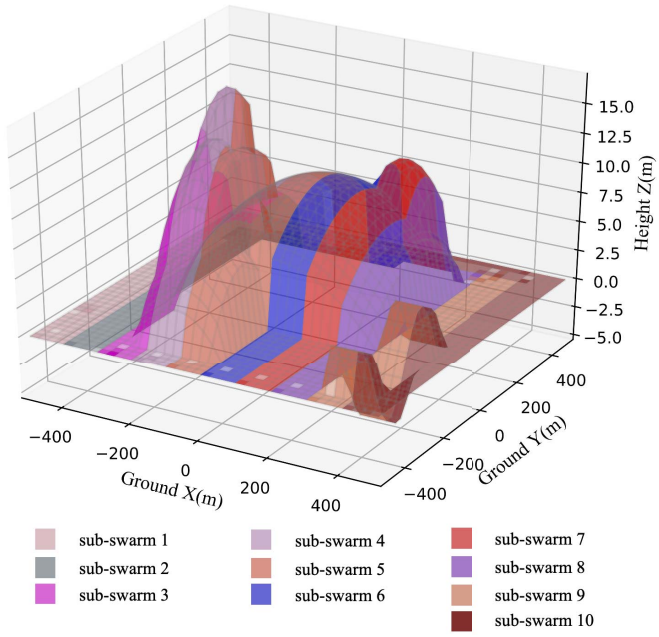


Fig. 13. 3D coverage result of UAV swarm using SDQN at 163,000 time steps when the UAV swarm finishes covering all the patches of the terrain, i.e., $r_a^c = 100\%$. Different colors represent covered patches by 10 different sub-swarms.

indicates that SDQN algorithm can guarantee the fairness in the cooperation between different sub-swarms. The repeated coverage rate r_p^c is $r_p^c = 2.71\% < 5\%$, which indicates that SDQN algorithm satisfies the repeated coverage constrains.

The 3D coverage of UAV swarm using SDQN algorithm at 163,000 time step is shown in Fig. 13, where different colors represent the patches covered by different sub-swarms. The sub-swarms cover the patches within their lanes and help other sub-swarms that have not complete the coverage of their own lanes yet. Finally, the sub-swarms cooperatively cover the entire areas of terrain $\mathcal{Q}$. The connectivity of LCN during coverage task is examined by DFS algorithm. The LCN remains connected at every time step during the flight, which satisfies the connectivity constrains of the coverage problem.

VII. Conclusion

In this paper, we considered a 3D terrain surface coverage problem with hierarchical UAV swarm. We constructed a 3D terrain model using regular grid method and projected the terrain into 2D patches. We also developed a patch area calculation algorithm to build a hotmap of the terrain that could determine the time consumption to cover each patch. The UAV swarm is designed to have a two-level hierarchy structure, where the high-level LUAVs were responsible for selecting patches and the low-level FUAVs carried out the specific coverage within patches. For FUAVs, we designed the coverage trajectory based on the star communication topology within sub-swarms. For LUAVs, we proposed a swarm DRL algorithm, SDQN, for the patch selection problem. SDQN can address the partial observation issue of LUAVs, and has low communication overloads. The simulations showed that the proposed coverage trajectory algorithm was able to make FUAVs cover all the areas of the patch with little redundancies. The total coverage time of SDQN algorithm was less than that of existing algorithms, including VPM, MSTC, and other reinforcement learning algorithms. Moreover, SDQN has a low repeated coverage rate and can guarantee the fairness among sub-swarms. The algorithms we proposed is suitable for the scenarios where UAVs need to fly close to the 3D terrain surface to cover all the areas in detail for detections, collections or other purposes.

Appendix
Upper Bounds of Curve Length on Terrain

As shown in Fig. 4, we set up a new coordinate for edge $\overset{\frown}{AB}$ of patch $ABCD$ under the X-Y-Z frame, where s-axis coincides with line $\overline{AB}$. We express $\overset{\frown}{AB}$ as a function $f(s)$, $s \in [0, |\overline{AB}|]$. The maximum relative slope C_{ms} is defined as the supremacy of the derivation of $f(s)$, i.e.,

$$C_{ms} = \sup_s \frac{\mathrm{d}f(s)}{\mathrm{d}s}. \tag{65}$$

Based on the ideas of calculus, we divide $\overset{\frown}{AB}$ into many small segments $\mathrm{d}l$, where $\mathrm{d}l = \frac{\mathrm{d}s}{cos\theta}$, and its entire length can be expressed as

$$\begin{aligned} |\overset{\frown}{AB}| &= \int_0^{|\overline{AB}|} \mathrm{d}l = \int_0^{|\overline{AB}|} \frac{\mathrm{d}s}{cos\theta} \leq \frac{1}{cos(\theta_{max})} \int_0^{|\overline{AB}|} \mathrm{d}s \\ &= \frac{|\overline{AB}|}{cos(\theta_{max})}. \end{aligned} \tag{66}$$

Note that θ_{max} is the maximum slope angle related to C_{ms}, and Δy is the length of the rectangle edges defined in (2). Since $|\overline{AB}|$ has been derived in (19) and $\cos(\theta_{max})$ can be represented by $C_{ms} = \tan(\theta_{max})$, i.e., $\cos(\theta_{max}) = \sqrt{1 + C_{ms}^2}$, the length of $\overset{\frown}{AB}$ can be written as

$$|\overset{\frown}{AB}| \leq \sqrt{\frac{(\Delta y)^2 + (z_{i,j} - z_{i,(j+1)})^2}{1 + {C_{ms}}^2}}. \tag{67}$$

Other edges can be calculated in a similar way.

REFERENCES

[1] N. Zhao *et al.*, "UAV-assisted emergency networks in disasters," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 45–51, Feb. 2019.

[2] Y. Zhang, Z. Mou, F. Gao, L. Xing, J. Jiang, and Z. Han, "Hierarchical deep reinforcement learning for backscattering data collection with multiple UAVs," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3786–3800, Mar. 2021.

[3] G. Zhang, Q. Wu, M. Cui, and R. Zhang, "Securing UAV communications via trajectory optimization," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, Dec. 2017, pp. 1–6.

[4] Y. Zhang, Z. Mou, F. Gao, J. Jiang, R. Ding, and Z. Han, "UAV-enabled secure communications by multi-agent deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11599–11611, Oct. 2020.

[5] J. Zhao, F. Gao, Q. Wu, S. Jin, Y. Wu, and W. Jia, "Beam tracking for UAV mounted SatCom on-the-Move with massive antenna array," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 2, pp. 363–375, Feb. 2018.

[6] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in UAV enabled wireless sensor network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 328–331, Jun. 2018.

[7] Y. Gu, Y. Jiao, X. Xu, and Q. Yu, "Request–response and censoring-based energy-efficient decentralized change-point detection with IoT applications," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6771–6788, Apr. 2021.

[8] G. Yang, C. K. Ho, and Y. L. Guan, "Multi-antenna wireless energy transfer for backscatter communication systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2974–2987, Dec. 2015.

[9] 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.

[10] P. Zhao *et al.*, "Optimal trajectory planning of drones for 3D mobile sensing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.

[11] X. He *et al.*, "Towards 3D deployment of UAV base stations in uneven terrain," in *Proc. 27th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Hangzhou, China, Jul. 2018, pp. 1–9.

[12] P. Cheng, J. Keller, and V. Kumar, "Time-optimal UAV trajectory planning for 3D urban structure coverage," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nice, France, Sep. 2008, pp. 2750–2757.

[13] M. Thanou and A. Tzes, "Distributed visibility-based coverage using a swarm of UAVs in known 3D-terrains," in *Proc. 6th Int. Symp. Commun., Control Signal Process. (ISCCSP)*, Athens, Greece, May 2014, pp. 425–428.

[14] Y. Kantaros, M. Thanou, and A. Tzes, "Visibility-oriented coverage control of mobile robotic networks on non-convex regions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Hong Kong, May 2014, pp. 1126–1131.

[15] G. Yang, Q. Zhang, and Y.-C. Liang, "Cooperative ambient backscatter communications for green Internet-of-Things," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1116–1130, Apr. 2018.

[16] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.

[17] X. Zhou, W. Wang, T. Wang, X. Li, and Z. Li, "A research framework on mission planning of the UAV swarm," in *Proc. 12th Syst. Syst. Eng. Conf. (SoSE)*, Waikoloa, HI, USA, Jun. 2017, pp. 1–6.

[18] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Perth, WA, Australia, Nov. 1995, pp. 1942–1948.

[20] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Appl. Math. Comput.*, vol. 214, no. 1, pp. 108–132, Aug. 2009.

[21] M. Hüttenrauch, S. Adrian, and G. Neumann, "Deep reinforcement learning for swarm systems," *J. Mach. Learn. Res.*, vol. 20, no. 54, pp. 1–31, 2019.

[22] H. Iima, Y. Kuroe, and S. Matsuda, "Swarm reinforcement learning method based on ant colony optimization," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Istanbul, Turkey, Oct. 2010, pp. 1726–1733.

[23] X. Chen, J. Tang, and S. Lao, "Review of unmanned aerial vehicle swarm communication architectures and routing protocols," *Appl. Sci.*, vol. 10, no. 10, p. 3661, May 2020.

[24] F. J. Aguilar, F. Agüera, M. A. Aguilar, and F. Carvajal, "Effects of terrain morphology, sampling density, and interpolation methods on grid DEM accuracy," *Photogrammetric Eng. Remote Sens.*, vol. 71, no. 7, pp. 805–816, Jul. 2005.

[25] G. H. Schut, "Review of interpolation methods for digital terrain models," *Can. Surveyor*, vol. 30, no. 5, pp. 389–412, Dec. 1976.

[26] N. Goddemeier and C. Wietfeld, "Investigation of air-to-air channel characteristics and a UAV specific extension to the rice model," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, San Diego, CA, USA, Dec. 2015, pp. 1–5.

[27] M. J. McCullagh, "Terrain and surface modelling systems: Theory and practice," *Photogramm. Rec.*, vol. 12, no. 72, pp. 747–779, Aug. 2006.

[28] Y. Qian, P. Cao, W. Yin, F. Dai, F. Hu, and Z. Yan, "Calculation method of surface shape feature of rice seed based on point cloud," *Compt. Electron. Agric.*, vol. 142, pp. 416–423, Nov. 2017.

[29] J. F. Araujo, P. B. Sujit, and J. B. Sousa, "Multiple UAV area decomposition and coverage," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl. (CISDA)*, Singapore, Apr. 2013, pp. 30–37.

[30] L. H. Nam, L. Huang, X. J. Li, and J. F. Xu, "An approach for coverage path planning for UAVs," in *Proc. IEEE 14th Int. Workshop Adv. Motion Control (AMC)*, Auckland, New Zealand, Apr. 2016, pp. 411–416.

[31] M. Torres, D. A. Pelta, J. L. Verdegay, and J. C. Torres, "Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction," *Expert Syst. Appl.*, vol. 55, pp. 441–451, Aug. 2016.

[32] Z. Galil and G. F. Italiano, "Data structures and algorithms for disjoint set union problems," *ACM Comput. Surveys*, vol. 23, no. 3, pp. 319–344, Sep. 1991.

[33] D. Chen, Q. Qi, Z. Zhuang, J. Wang, J. Liao, and Z. Han, "Mean field deep reinforcement learning for fair and efficient UAV control," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 813–828, Jan. 2021.

[34] R. Ding, F. Gao, and X. S. Shen, "3D UAV trajectory design and frequency band allocation for energy-efficient and fair communication: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 7796–7809, Dec. 2020.

[35] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*. Berlin, Germany: Springer, 2012.

[36] I. L. Glicksberg, "A further generalization of the Kakutani fixed point theorem, with application to Nash equilibrium points," *Proc. Amer. Math. Soc.*, vol. 3, no. 1, pp. 170–174, Feb. 1952.

[37] P. Sunehag *et al.*, "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proc. Int. Conf. Auton. Agents MultiAgent Syst. (AAMAS)*, Stockholm, Sweden, Jul. 2018, pp. 2085–2087.

[38] A. Smola, A. Gretton, L. Song, and B. Schölkopf, "A Hilbert space embedding for distributions," in *Proc. Int. Conf. Algorithm. Learn. Theory* Berlin, Germany: Springer, Oct. 2007, pp. 13–31.

[39] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, "QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, Long Beach, CA, USA, Jun. 2019, pp. 5887–5896.

[40] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[41] G. Yang, Y.-C. Liang, R. Zhang, and Y. Pei, "Modulation in the air: Backscatter communication over ambient OFDM carrier," *IEEE Trans. Commun.*, vol. 66, no. 3, pp. 1219–1233, Mar. 2018.

[42] E. Ventocilla, "Swarm-based area exploration and coverage based on pheromones and bird flocks," M.S. thesis, Dept. Inform. Media, Uppsala Univ., Uppsala, Sweden, 2013.

[43] N. Hazon and G. A. Kaminka, "On redundancy, efficiency, and robustness in coverage for multiple robots," *Robot. Auto. Syst.*, vol. 56, no. 12, pp. 1102–1114, Dec. 2008.

Zhiyu Mou received the B.Eng. degree in automation from the Beijing Institute of Technology, China, in 2020. He is currently pursuing the M.S. degree with the Department of Automation, Tsinghua University, Beijing, China. His research interests include deep reinforcement learning, optimizations, and swarm systems.

Yu Zhang (Graduate Student Member, IEEE) received the B.Eng. degree in communication engineering from Shandong University, China, in 2016. She is currently pursuing the Ph.D. degree with the Department of Automation, Tsinghua University, Beijing, China. She was a visiting Ph.D. student with the Department of Electrical and Computer Engineering, University of Houston in 2019. Her research interests include performance analysis, backscatter communications, optimization, deep reinforcement learning, and UAV.

Feifei Gao (Fellow, IEEE) received the B.Eng. degree from Xi'an Jiaotong University, Xi'an, China, in 2002, the M.Sc. degree from McMaster University, Hamilton, ON, Canada, in 2004, and the Ph.D. degree from the National University of Singapore, Singapore, in 2007.

Since 2011, he has been with the Department of Automation, Tsinghua University, Beijing, China, where he is currently an Associate Professor. He has authored/coauthored more than 150 refereed IEEE journal articles and more than 150 IEEE conference proceeding articles that are cited more than 8800 times in Google Scholar. His research interests include signal processing for communications, array signal processing, convex optimizations, and artificial intelligence assisted communications. He has served as the Symposium Co-Chair for IEEE Conference on Communications (ICC) in 2019, IEEE Vehicular Technology Conference Spring (VTC) in 2018, IEEE Conference on Communications (ICC) in 2015, IEEE Global Communications Conference (GLOBECOM) in 2014, IEEE Vehicular Technology Conference Fall (VTC) 2014, as well as Technical Committee Member for more than 50 IEEE conferences. He has also served as an Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, a Lead Guest Editor for IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, and a Senior Editor for IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE SIGNAL PROCESSING LETTERS, IEEE COMMUNICATIONS LETTERS, IEEE WIRELESS COMMUNICATIONS LETTERS, and *China Communications*.

Huangang Wang received the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China, in 2005. He is an Associate Professor with the Department of Automation, Tsinghua University. His research interests include nonlinear control, optimization of manufacturing systems, machine learning, and intelligent control systems.

Tao Zhang (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Tsinghua University, Beijing, China, in 1993, 1995, and 1999, respectively, and the Ph.D. degree from Saga University, Saga, Japan, in 2002. He is currently a Professor and the Deputy Head of the Department of Automation, School of Information Science and Technology, Tsinghua University. He is the author or coauthor of more than 200 articles and 3 books. His current research includes robotics, image processing, control theory, artificial intelligent, navigation, and control of spacecraft.

Zhu Han (Fellow, IEEE) received the B.S. degree in electronic engineering from Tsinghua University in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively.

From 2000 to 2002, he was a Research and Development Engineer with JDSU, Germantown, MD, USA. From 2003 to 2006, he was a Research Associate with the University of Maryland. From 2006 to 2008, he was an Assistant Professor with Boise State University, Boise, ID, USA. He is currently a John and Rebecca Moores Professor with the Electrical and Computer Engineering Department and with the Computer Science Department, University of Houston, Houston, TX, USA. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. Since 2019, he has been an AAAS Fellow and an ACM Distinguished Member. He received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the *Journal on Advances in Signal Processing* in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (Best Paper Award in IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS) in 2016, and several best paper awards in IEEE conferences. He was an IEEE Communications Society Distinguished Lecturer from 2015 to 2018. He has been 1% highly cited researcher since 2017 according to Web of Science. He is also the winner of the IEEE Kiyo Tomiyasu Award in 2021, for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: "for contributions to game theory and distributed management of autonomous communication networks."