Multi-UAVs Strategies for Ad Hoc Network with Multi-Agent Reinforcement Learning

George Karimata*, Jin Nakazato*, Gia Khanh Tran†, Katsuya Suto‡, Manabu Tsukada* and Hiroshi Esaki*
*The University of Tokyo, Japan
†Tokyo Institute of Technology, Japan,
‡The University of Electro-Communications, Japan
Email: {karimata-george030, jin-nakazato, mtsukada}@g.ecc.u-tokyo.ac.jp

***Abstract*—In recent years, extensive research has focused on leveraging advanced technologies beyond 5G and for Industry 5.0 to promote sustainability and prosperity in society. Our study advances this effort by seeking to create an aerial perspective using Unmanned Aerial Vehicles (UAVs). This paper introduces a method for optimizing UAV deployment strategies using multi-agent reinforcement learning, facilitating the formation of a flying ad hoc network. The results demonstrate practical cooperation among UAVs in flight.**

***Index Terms*—UAV, flying ad hoc network, multi-agent reinforcement learning, multi-agent transformer**

I. Introduction

Recently, Industry 5.0 has been envisioned by the European Commission as a roadmap for the future, aiming to cultivate a prosperous society through the strategic use of advanced technologies such as Artificial Intelligence (AI), Robotics, Big Data, and the Internet of Things (IoT) [1]. These technologies find practical applications in Digital Twins (DT), which serve as the conduit between the real and virtual worlds [2]. Unmanned Aerial Vehicles (UAVs), capable of freely moving in three dimensions, are instrumental in realizing Industry 5.0. These UAVs are slated for deployment across a range of sectors. Furthermore, recent research has shed light on the capabilities of Non-Terrestrial Networks (NTNs) to equip UAVs and Low Earth Orbit satellites with robust communication systems. These networks are precious when terrestrial infrastructure is compromised by natural disasters such as earthquakes and typhoons [3]. Among the promising solutions in this arena are UAV base stations (UAV-BS), which offer the flexibility to adjust their positions rapidly. These stations are increasingly seen as vital for enhancing communication performance in areas experiencing sudden spikes in data traffic or regions affected by disasters [4].

There are many studies on UAV deployment and flight routing challenges. Park et al. proposed an algorithm to move UAVs according to the distance between UAVs and the number of mobile devices connected to them [5]. It has been successful in properly positioning the UAV swarm for stationary users and increasing communication capacity. Bayerlein et al. used multi-agent reinforcement learning to efficiently navigate multiple UAVs for data collection from IoT sensors, navigating around no-fly zones and within limited flight distances [6].

However, many studies, including those mentioned above, focus on the optimal deployment of UAVs for static users and flight routing without considering communication range constraints. This paper presents a novel approach by addressing the deployment and routing issues for moving UAVs within an ad hoc network subject to communication distance limitations.

The goal of this paper is to derive an optimal strategy for determining the flight paths of a UAV smarm using Multi-Agent Reinforcement Learning (MARL), a proven solution for cooperative tasks. Given the dynamic nature of the scenario—where multiple UAVs must navigate toward destinations within an ad hoc network—collaboration is imperative. MARL has been extensively studied; for instance, Multi-Agent Proximal Policy Optimization (MAPPO) [7] extends the single-agent Proximal Policy Optimization (PPO) [8] algorithm by incorporating other agents' data into value calculations. In contrast, Heterogeneous-Agent Proximal Policy Optimization (HAPPO) [9] addresses the limitations of MAPPO by accommodating agents with varying action spaces. Wen et al. conceptualized MARL through a sequential model. They introduced a Multi-Agent Transformer (MAT) [10], as depicted in Fig. 1, combined with the Transformer architecture [11], known for its efficacy with sequential data, such as natural language. In this paper, simulations were performed using MAT. The primary contributions of this paper are as follows.

- Identify flight routes to destinations that maintain a functional ad hoc network, considering communication range constraints.
- Coordinate the movement of UAVs to preserve the integrity of the ad hoc network through strategic positioning.

The remainder of this paper is organized as follows. Section Ⅱ introduces our proposed methodology, Section Ⅲ discusses the simulation setup and results. Finally, Section Ⅳ concludes this paper.

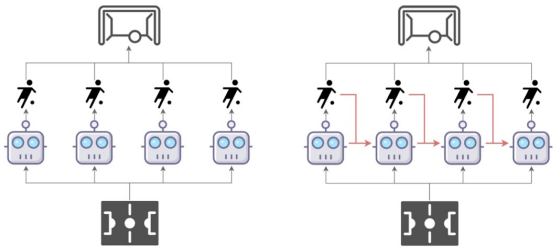

Fig. 1. Conventional multi-agent learning paradigm (left) and the multi-agent sequential decision paradigm (right) [10]

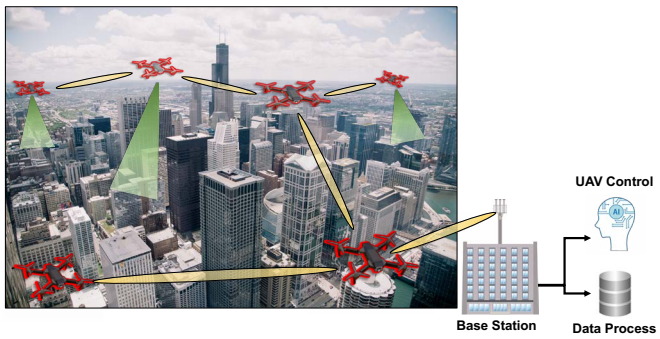

Fig. 2. Proposed system model

II. System Design

A. Design of Reward Function

The following description summarizes the settings assumed in the proposed system model, as illustrated in Figure 2. Within this model, a group of UAVs forms an ad hoc network. The UAVs are initially located at the BS and move around the target area to collect data from specific destinations, referred to as Landmarks. The UAV can obtain information such as the position and velocity of all UAVs via an ad hoc network.

- There is a limit to the reach of radio waves, and let L_{wave} be the distance.
- Initial positions of UAVs are the same as BS.
- Set of UAVs is $\boldsymbol{U}$ and $\boldsymbol{U} = \{u_1, \ldots, u_n\}$.
- Set of Landmarks is $\boldsymbol{L}$ and $\boldsymbol{L} = \{l_1, \ldots, l_m\}$.

The conditions to be met are as follows.

1) At least one UAV should reach each Landmark.
2) UAVs should be reasonably far from each other to minimize radio interference.
3) Ensure the connectivity of ad hoc network.
4) The number of flying UAVs should be minimized to satisfy the above conditions.

The rewards should be set such that these four conditions are met. We set the reward that a single UAV will receive for each condition. Hereinafter, dist$(Entity_1, Entity_2)$ shall represent the distance between $Entity_1$ and $Entity_2$.

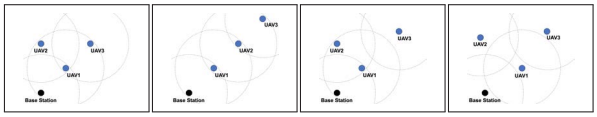

Fig. 3. Positional relationship of four cases

(1) The reward R_l involved in Landmarks in Condition 1 is formulated as follows.

$$R_l = \sum_{l_i \in \boldsymbol{L}} r_{l_i}$$

$$r_{l_i} = \begin{cases} 0 & \text{(if any UAV has already visited } l_i\text{)} \\ -\min\limits_{u_j \in \boldsymbol{U}} (\text{dist}(l_i, u_j)) & \text{(else)} \end{cases}$$

If any UAV has visited a Landmark, no penalty is applied to the reward for that Landmark. However, if no UAV has reached a particular Landmark, the distance from that Landmark to the nearest UAV is deducted from the reward. This system enhances the reward when a UAV approaches an unreached Landmark, thus preventing the scenario where all UAVs converge on the same Landmark.

(2) The reward R_u associated with the UAV in Condition 2 is determined by itself and the UAV $u_{neighber}$ closest to itself.

$$R_u = \begin{cases} 0 & \text{(if } itself \text{ is at BS)} \\ r_u & \text{(else)} \end{cases}$$

$$r_u = \begin{cases} \min\left(0, L_{wave} - \text{dist}(itself, u_{neighber})\right) \\ \qquad\qquad \text{(if } u_{neighber} \text{ is at BS)} \\ -\left|L_{wave} - \text{dist}(itself, u_{neighber})\right| \\ \qquad\qquad \text{(else)} \end{cases}$$

When the UAV is positioned at BS, it should not accrue any rewards for maintaining its position there. However, when a UAV is located outside BS, its reward system will be influenced by the proximity of other UAVs. If $u_{neighber}$ is at BS and the distance is within radio range L_{wave}, no penalty is given. The reason is that if a penalty is given, once the UAV leaves BS, it will leave BS by L_{wave} and will have difficulty approaching the Landmarks in the vicinity of BS. On the other hand, when $u_{neighber}$ is at BS, and the distance is out of radio range, or when $u_{neighber}$ is outside BS, penalize the difference between L_{wave} and the distance to $u_{neighber}$ to keep the distance at L_{wave}. These settings are expected to separate the UAVs by L_{wave}, especially between UAVs outside BS, and to distribute the UAVs.

(3) The connectivity reward R_c in Condition 3 is determined using an adjacency matrix, which is set to 1 when the distance between BS and UAV is less than L_{wave} and 0 when the distance is greater than L_{wave}. When the adjacency matrix is multiplied by the number of UAVs n, it can be determined that the connection is possible when all components are greater than or equal to 1. For example, when BS and three UAVs are in the positional relationship of four cases, as shown in Fig. 3,

the adjacency matrix is $A_1,\ A_2,\ A_3,\ A_4$ from left to right. The powers of each are as follows.

$$A_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad (A_1)^2 = \begin{bmatrix} 2 & 2 & 1 & 1 \\ 2 & 4 & 2 & 2 \\ 1 & 2 & 2 & 1 \\ 1 & 2 & 1 & 2 \end{bmatrix} \quad (A_1)^3 = \begin{bmatrix} 4 & 6 & 3 & 3 \\ 6 & 10 & 6 & 6 \\ 3 & 6 & 4 & 3 \\ 3 & 6 & 3 & 4 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad (A_2)^2 = \begin{bmatrix} 2 & 2 & 1 & 0 \\ 2 & 3 & 2 & 1 \\ 1 & 2 & 3 & 2 \\ 0 & 1 & 2 & 2 \end{bmatrix} \quad (A_2)^3 = \begin{bmatrix} 4 & 5 & 3 & 1 \\ 5 & 7 & 6 & 3 \\ 3 & 6 & 7 & 5 \\ 1 & 3 & 5 & 4 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (A_3)^2 = \begin{bmatrix} 2 & 2 & 1 & 0 \\ 2 & 3 & 2 & 0 \\ 1 & 2 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (A_3)^3 = \begin{bmatrix} 4 & 5 & 3 & 0 \\ 5 & 7 & 5 & 0 \\ 3 & 5 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (A_4)^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (A_4)^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We know that communication can hop from BS to all UAVs if all components of the adjacency matrix multiplied by n are greater than or equal to 1. Using this, the reward R_c is formulated as follows. The number of 0 elements and all elements of the adjacency squared to the power of n are defined as N_{zero} and N_{all}, respectively.

$$R_c = \frac{-N_{zero}}{N_{all} - n - 1} = \frac{-N_{zero}}{n(n+1)}$$

Since the diagonal component is always greater than or equal to 1, the reward is subtracted according to the amount of zeros in the other components.

(4) Finally, set the reward R_b related to BS in Condition 4.

$$R_b = \begin{cases} 0 & \text{(if the UAV is at BS)} \\ -1 & \text{(else)} \end{cases}$$

If not at BS, the penalty will be given. The reward earned by a single UAV is determined by adding up these four rewards (R_l, R_u, R_c, R_b) with their respective weights (w_l, w_u, w_c, w_b).

$$R = w_l R_l + w_u R_u + w_c R_c + w_b R_b \quad (w_l = w_u = 1)$$

In this paper, since we consider that the UAV swarm is going to Landmarks while maintaining the ad hoc network, we set the weight w_c of the reward R_c to ensure connectivity is sufficiently large. Maximal at $R_c = 0$ when all UAVs are connectable. The next largest value that can be taken is $R_c = \frac{-2n}{n(n+1)} = \frac{-2}{n+1}$ when only one UAV is isolated, as in $(A_3)^3$. R_l accounts for the largest portion of R, and its lowest value is suppressed from below by roughly $-\max_{l_i \in \boldsymbol{L}}(\text{dist}(BS, l_i)) * m \ (= -limit)$. Therefore, we set $w_c = limit * \frac{n+1}{2}$ to increase the penalty for connectivity over other rewards if even one UAV is isolated and fails to ensure connectivity. Also, by having one UAV relay away from BS, the UAV group can be expected to be approximately L_{wave} closer to the Landmark and R_l larger. The weight of R_b is $w_b = L_{wave}$.

TABLE I
HARDWARE SPECIFICATIONS

OS	Windows 11
CPU	12th Gen Intel(R) Core(TM) i9-12900HX 2.30GHz
GPU	NVIDIA GeForce RTX 4080 Laptop GPU
MEM	64GB
SSD	1TB

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Total Steps	128,000,000
Total Steps for Episode	100
Working Threads	256
Episodes	5,000
UAVs n	6
Landmarks m	2
Radio Wave Distance L_{wave}	5.0
Acceleration Vector Magnitude a	0.50
Area Range for Landmarks $limit$	13.0

B. MARL Training

The objective of this study is to employ MARL to devise a methodology for determining the flight paths of a group of UAVs that collectively form an ad hoc network. We conducted MARL with UAVs as agents to acquire strategies. We utilize the MAT within our framework. The training protocol proceeds as follows: At the start of each episode, the environment is initialized. Subsequently, the policy dictates the agents' actions, which are informed by the collective observed data and the preceding actions of other agents. Adhering to the principles of a Markov process, the forthcoming state is contingent on the current actions and state, prompting an update in the system. Upon updating, the reward for each agent is computed. This sequence of steps is iteratively conducted throughout the episode. Upon its conclusion, the policy is revised to reflect the cumulative observations, actions, and rewards acquired for the episode.

III. NUMERICAL RESULTS

A. Setup Conditions

The hardware specifications used in this study are summarized in Table I. The main software used was Python 3.11.6 and PyTorch 2.0.1+cu118. The simulations were performed using the publicly available training model MAT [12] and the Multi-Agent Particle Environment (MPE) [13] environment, modified to fit this study. The five available actions of agents are to accelerate with a magnitude a in either the x-y positive or negative direction or not to accelerate. The following equations update the agent's velocity $\boldsymbol{v}$ and position $\boldsymbol{p}$. The decay rate $\gamma = 0.75$ and the microtome $dt = 0.1$ are set.

$$\boldsymbol{v}_{current} = \boldsymbol{v}_{previous} * \gamma + \boldsymbol{a} * dt$$
$$\boldsymbol{p}_{current} = \boldsymbol{p}_{previous} + \boldsymbol{v}_{current} * dt$$

When initializing the environment for each episode, Landmarks are placed uniformly at random within a $2{*}limit$ square

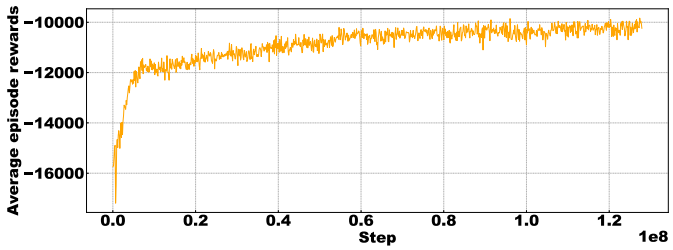

Fig. 4. Average episode rewards

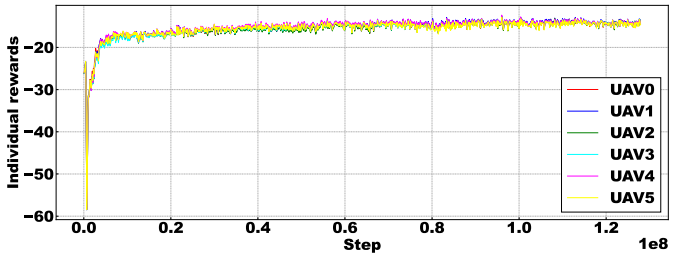

Fig. 5. Individual rewards

centered at BS. The simulation parameters are summarized in Table II.

B. Simulation Results

The average episode rewards, as depicted in Fig. 4, along with the individual rewards of agents, illustrated in Fig. 5, both exhibit an upward trend with the progression of learning. Sequential snapshots taken at steps 25, 50, 75, and 100 within episode 5,000, presented in Fig. 6, demonstrate that the UAVs can move while preserving network connectivity. These images further reveal the strategic movements of the UAVs. Some reach or are near Landmarks, while others strategically maintain a certain distance from the BS. These movements collectively suggest coordinated behavior among UAVs.

IV. Conclusion

In this study, we proposed a method for deriving deployment strategies for UAV groups that form an ad hoc network, utilizing multi-agent reinforcement learning with MAT. Simulations were conducted with reward structures designed to foster the development of an ad hoc network. The results indicated that, as learning progressed, the UAVs increasingly exhibited cooperative flight behavior. Future work will involve adopting an environmental model that more closely mirrors real-world conditions, incorporating factors such as radio interference. We aim to assess the effectiveness of our proposed method further by exploring the optimization of routing paths through a routing protocol and comparing the performance of the ad hoc network against alternative approaches.

Acknowledgment

This research was supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE) by the Ministry of Internal Affairs and Communications, Japan (receipt number JP235003015).

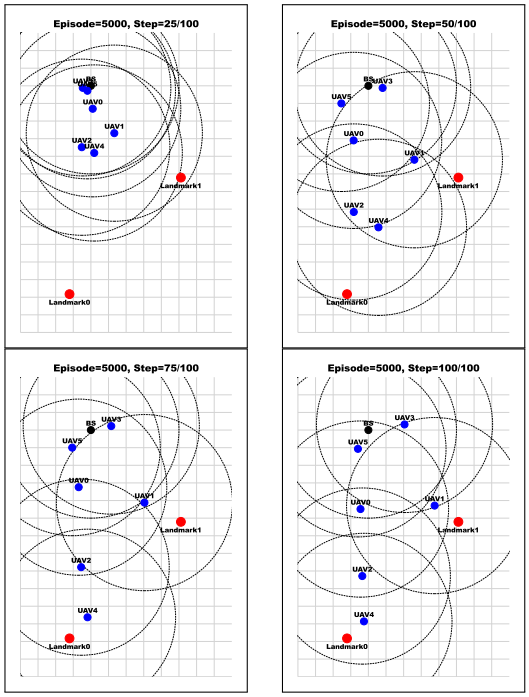

Fig. 6. Snapshots of each situation in episode 5,000

References

[1] S. K. Jagatheesaperumal and M. Rahouti. Building digital twins of cyber physical systems with metaverse for industry 5.0 and beyond. *IT Professional*, 24(6):34–40, 2022.

[2] K. Inokuchi et al. Semantic digital twin for interoperability and comprehensive management of data assets. In *2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom)*, pages 217–225, 2023.

[3] M. M. Azari and thers. Evolution of non-terrestrial networks from 5g to 6g: A survey. *IEEE Communications Surveys & Tutorials*, 24(4):2633–2672, 2022.

[4] G. K. Tran et al. NFV/SDN as an Enabler for Dynamic Placement Method of mmwave Embedded UAV Access Base Stations. *MDPI Network*, 2022.

[5] S. Park et al. Formation control algorithm of multi-uav-based network infrastructure. *Applied Sciences*, 8(10), 2018.

[6] H. Bayerlein et al. Multi-uav path planning for wireless data harvesting with deep reinforcement learning. *IEEE Open Journal of the Communications Society*, 2:1171–1187, 2021.

[7] C. Yu et al. The surprising effectiveness of ppo in cooperative multi-agent games. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24611–24624. Curran Associates, Inc., 2022.

[8] J. Schulman et al. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[9] J. G. Kuba et al. Trust region policy optimisation in multi-agent reinforcement learning. *CoRR*, abs/2109.11251, 2021.

[10] M. Wen et al. Multi-agent reinforcement learning is a sequence modeling problem. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 16509–16521. Curran Associates, Inc., 2022.

[11] A. Vaswani et al. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[12] https://github.com/PKU-MARL/Multi-Agent-Transformer (Accessed on 2023).

[13] https://github.com/openai/multiagent-particle-envs/ (Accessed on 2023).