Cooperative UAVs with Asynchronous Multi-agent Learning for Remote Data Collection

Cuong Le, Thang X. Vu, *Senior Member, IEEE*, Symeon Chatzinotas, *Fellow, IEEE*

Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg

***Abstract*—This paper addresses the optimization of trajectories for multiple Unmanned Aerial Vehicles (UAVs) and bandwidth allocation to enhance energy efficiency in a general cooperative data collection problem. We focus on a practical scenario where sensor nodes (SNs) are distributed over a remote area without terrestrial infrastructures and UAVs have to make their decisions asynchronously based on local information, with inter-UAV information exchange only possible when they are within communication range. While bandwidth allocation can be solved based on local observation at each hovering point, trajectory planning is not as straightforward due to imperfect information and asynchronous decisions. To tackle these challenges, we formulate the trajectory planning problem as a Decentralized Partially Observable Semi-Markov Decision Process (Dec-POSMDP), for which we introduce an asynchronous version of a well-known algorithm called QMIX to learn UAVs' policies. We also provide empirical evidences to demonstrate the learning performance of the proposed method, as well as the robustness of the learned policies in response to varying UAV configurations.**

***Index Terms*—Data collection, unmanned aerial vehicles, multi-agent reinforcement learning, cooperative multi-agent systems.**

I. Introduction

In IoT systems deployed in remote areas, numerous sensor nodes (SNs) are placed to monitor environmental parameters such as temperature, humidity, air quality, and wildlife activity. Due to the vast and often inaccessible terrain of these areas, deploying central base stations for data retrieval can be challenging and cost-inefficient. To overcome these limitations, deploying unmanned aerial vehicles (UAVs) is a potential solution thanks to their accessibility and highly probable line-of-sight links towards the SNs. However, safety requirements and efficient cooperative operation under limited energy remain challenges. Overcoming these hurdles is essential to maximize the potentials of UAVs and ensuring sustainable use.

Early research in UAVs-aided data collection start with simple scenarios, with static and deterministic assumptions, allowing traditional optimization approaches to be applied [1], [2]. As investigated systems become increasingly intricate, offline optimization exposes limitations. Instead, reinforcement learning, with its adaptability to complex and dynamic environments, on the other hand, provides the flexibility needed to cope with practical scenarios [3], [4]. Despite the attainment of some promising results, there are still technical gaps separating these studies from practical systems. Particularly, studies often assume deterministic systems where either perfect channel state information (CSI) or data collection demand are known in advance at every possible UAV locations. Meanwhile, learning-based studies often assume that the central controller and UAVs have the capability to observe the entire collection area in real-time [4]. While these assumptions may facilitate the applications of optimization and learning algorithms, they are not in line with practical scenarios, specifically in the context of remote data collection.

Some of the most recent studies have made efforts to tackle the partial observable challenges, where the problem is often formulated as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP), to which a multi-agent reinforcement learning algorithm is applied [5], [6]. Although offering fairly comprehensive solutions, these studies still heavily rely on a critical assumption regarding the synchronization of UAVs for the convenience of mathematical modeling and problem-solving. Specifically, the time evolution is discritized into equally small intervals, and UAVs' decisions are made in each interval at the same time. This assumption, however, is infeasible in real-world scenarios, especially in such large remote areas with limited inter-UAV communication. Moreover, synchronous decision-making is time-inefficient, as synchronization necessitates unnecessary waiting periods due to the varying nature of decision epochs among different UAVs. Furthermore, discretizing the timeline as said can be theoretically problematic, as decision-making under Dec-POMDP is itself NEXP-complete, and the complexity grows double-exponentially in the planning horizon [7]. Besides, impacts of inter-UAV communication have been overlooked in existing studies.

In this paper, we fill the aforementioned research gaps by focusing on a general data collection problem under practical constraints. The key novelty of our framework lies in UAVs autonomously making decisions asynchronously based on local information, without synchronizing with the actions of other UAVs. Our goal is to jointly optimize UAVs' trajectories and bandwidth allocation at each hovering point to enhance overall energy efficiency, while assuming that UAVs can only observe data collection demands in local regions around their positions and cannot estimate the CSI until they approach SNs. Moreover, to enhance cooperation, UAVs are able to exchange their operational histories if they are within their communication range. Given insufficient information available, we develop a real-time and autonomous solution where UAVs can make decisions on-trip. To this end, we decompose the problem into two sub-problems, including trajectory optimization and bandwidth allocation. While the latter can be

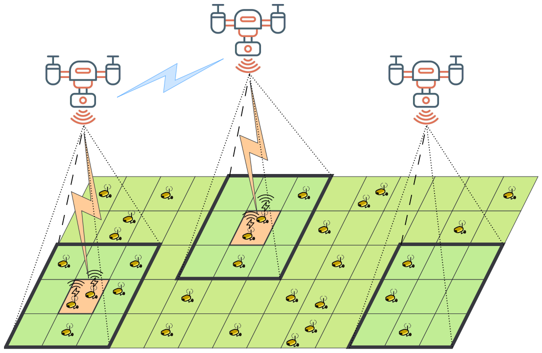


Fig. 1: Illustration of the investigated system. Blue links represent inter-UAV communication, and orange links represent SN-UAV data transmission. Cells with dark green and bold borders under UAVs represent their observable regions.

handled by a standard method, the former is reformulated as a decision-making problem under the Decentralized Partially Observable Semi-Markov Decision Process (Dec-POSMDP) [8]. We then introduce an asynchronous version for the QMIX algorithm [9], designed specifically for asynchronous environments. Experiment results suggest that the proposed method not just successfully overcomes the asynchronous challenge but also achieves highly competitive performance compared to other baselines.

II. System Model and Problem Formulation

There are N rotary-wing UAVs (not necessarily homogeneous) indexed by $\mathcal{N} = \{1, 2, \dots, N\}$ where UAV n departs from an initial location $\mathbf{w}_0^n$, travels around the monitored area to explore and collect data from SNs, and then lands at a final destination $\mathbf{w}_\text{F}^n$. The monitored area is divided into a grid of equal-size cells indexed by $\mathcal{C} = \{1, 2, \dots, H \times H\}$. Let $\mathcal{Q}_\text{C} = \{\mathbf{q}_\text{C}^c \in \mathbb{R}^2 : c \in \mathcal{C}\}$ be the set of all cell centers. Let $\mathcal{I} = \{1, 2, \dots, I\}$ be the set of I SNs, where SN i is located at a fixed location $\mathbf{q}_\text{SN}^i \in \mathbb{R}^2$. The availability of data at each SN follows a random process, where SN i has a data size of D^i bits to transmit if it is activated. UAVs can move between cells' centers to explore data collection demands, or can hover above these points to collect data. When a UAV decides to hover to collect data from a cell, it only leaves after collecting all available data. Inter-UAV communication is only possible within their communication range. We also define the termination conditions for a UAV, wherein it concludes the task and flies to its final destination either upon running out of energy or upon verifying that all data have been collected. Moreover, the termination of each UAV does not affect the operations of others. The investigated system is illustrated in Fig. 1.

A. UAVs' trajectories and energy consumption

Let $\mathcal{T}^n = \{t^n[0], t^n[1], \dots, t^n[K^n]\}$ represent the instants of time at which the n-th UAV makes decisions in its flight, and $\mathcal{K}^n = \{0, 1, \dots, K^n\}$ be the indices of these decisions. Assume that all UAVs operate at the same fixed altitude h. Let $\mathbf{w}^n = \{\mathbf{w}^n[0], \mathbf{w}^n[1], \dots, \mathbf{w}^n[K^n+1]\}$ be the corresponding trajectory of the n-th UAV, where $\mathbf{w}^n[k] \in \mathcal{Q}_\text{C}$ is the projection on the ground of the UAV's position at time $t^n[k]$. We have the first constraints regarding to the departure and arrival locations of UAVs given by

$$\mathbf{w}^n[0] = \mathbf{w}_0^n, \ \ \mathbf{w}^n[K^n+1] = \mathbf{w}_\text{F}^n, \forall n \in \mathcal{N}. \tag{1}$$

Let $\tau_\text{H}^n[k]$ represent the amount of time the n-th UAV hovers above $\mathbf{w}^n[k]$, and $\tau_\text{F}^n[k]$ represent the amount of time required for this UAV to fly from $\mathbf{w}^n[k]$ to $\mathbf{w}^n[k+1]$. If, at time $t^n[k]$, the UAV decides to hover above its current location $\mathbf{w}^n[k]$ to collect data, we have $\mathbf{w}^n[k] = \mathbf{w}^n[k+1]$ and $\tau_\text{F}^n[k] = 0$. Otherwise, we have $\tau_\text{H}^n[k] = 0$ and $\tau_\text{F}^n[k] = \frac{\|\mathbf{w}^n[k+1] - \mathbf{w}^n[k]\|_2}{v^n}$, where v^n is the fixed velocity of the n-th UAV. The total operating time of the n-th UAV can then be calculated by $T^n = \sum_{k=0}^{K^n} \tau_\text{F}^n[k] + \sum_{k=0}^{K^n+1} \tau_\text{H}^n[k]$, and the mission completion time by $T = \max_{n \in \mathcal{N}} T^n$.

The n-th UAV is powered by an on-board battery with limited capacity of E_max^n. Since the communication energy is negligible compared to that required for propulsion [10], we ignore this component in our analyses. Let $P_\text{UAV}(v^n)$ be the propulsion power consumption at velocity v^n of the n-th UAV, which can be calculated following a model in [10]. Let $E^n[k]$ be the remaining energy of this UAV at the time $t^n[k]$. Assume that all UAVs start with full batteries, i.e., $E^n[0] = E_\text{max}^n$. The remaining energy $E^n[k]$ at $t^n[k]$ can be calculated as $E^n[k] = E_\text{max}^n - P_\text{UAV}(0) \sum_{k'=0}^{k-1} \tau_\text{H}^n[k'] - P_\text{UAV}(v^n) \sum_{k'=0}^{k-1} \tau_\text{F}^n[k']$. To ensure safety during the mission, we impose the following constraints (for all n, k) to guarantee that UAVs always have sufficient energy to reach their final destinations

$$E_\text{max}^n - E^n[k] - P_\text{UAV}(0)\tau_\text{H}^n[k] \geq \xi^n(\mathbf{w}^n[k]) + \epsilon \tag{2}$$

$$E_\text{max}^n - E^n[k] - P_\text{UAV}(v^n)\tau_\text{F}^n[k] \geq \xi^n(\mathbf{w}^n[k+1]) + \epsilon \tag{3}$$

where $\xi^n(\mathbf{x}) = \frac{P_\text{UAV}(v)\|\mathbf{x} - \mathbf{w}_\text{F}^n\|_2}{v^n}$ is the energy required to fly to the final destination $\mathbf{w}_\text{F}^n$ from $\mathbf{x}$, and ϵ is a small safe energy margin. Let $\mathbf{w} = [\mathbf{w}^1, \mathbf{w}^2, \dots, \mathbf{w}^N]$ be the trajectories of all UAVs and $\mathbf{b}$ be their bandwidth allocation strategies (which will be defined more details in the following subsection). The total energy consumed by all UAVs can be calculated by

$$\psi(\mathbf{w},\mathbf{b}) = \sum_{n \in \mathcal{N}} \sum_{k \in \mathcal{K}^n} \left(P_\text{UAV}(0)\tau_\text{H}^n[k] + P_\text{UAV}(v^n)\tau_\text{F}^n[k]\right). \tag{4}$$

B. Data transmission model

Let $h^{in}(t)$ be the uplink channel between the n-th UAV and the i-th SN at the time t. This channel can be modeled as $h^{in}(t) = \sqrt{\alpha^{in}(t)} g^{in}(t)$ where $\alpha^{in}(t)$ is large-scale fading channel power gain due to pathloss and shadowing and $g^{in}(t)$ is small-scale fading coefficient. Let $P_{LoS}^{in}(t)$ and $P_{NLoS}^{in}(t)$ be the occurrence probabilities of line-of-sight (LoS) and non-line-of-sight (NLoS) connectivities between UAV n and SN i at the time t, respectively. These probabilities can be calculated following an approximation in [11]. The channel power gain due to large-scale fading can then be modeled as

$$\alpha^{in}(t) = \begin{cases} \alpha_0 \left(d^{in}(t)\right)^{-\eta}, & \text{w.p. } P_{LoS}^{in}(t) \\ \alpha_0 \beta \left(d^{in}(t)\right)^{-\eta}, & \text{otherwise} \end{cases} \tag{5}$$

where $\eta \geq 2$ is the pathloss exponent, β is the attenuation due to NLoS, $d^{in}(t) = \sqrt{\|\mathbf{w}^n(t) - \mathbf{q}^i_{\mathrm{SN}}\|_2^2 + h^2}$ is the distance between the i-th SN and the n-th UAV at time t, and $\alpha_0 = (4\pi f_c/c)^{-2}$ is the free-space channel power gain at distance of 1m with f_c being the carrier frequency and c being the speed of light. Since the UAVs often operate at high altitudes, LoS links are likely to occur, and thus, Rician fading is utilized to model the contribution of small-scale fading as $g^{in}(t) = \bar{g}^{in}\sqrt{\kappa^{in}(t)/(\kappa^{in}(t)+1)} + \hat{g}^{in}\sqrt{1/(\kappa^{in}(t)+1)}$, where $\bar{g}^{in}$ is the deterministic LoS component with $|\bar{g}^{in}| = 1$, $\hat{g}^{in} \sim \mathcal{CN}(0,1)$ represents all random scattered paths, $\kappa^{in}(t)$ is the Rician-factor that depends on the elevation angle $\theta^{in}(t)$ as $\kappa^{in}(t) = A_1 \exp\left(A_2\theta^{in}(t)\right)$, with A_1 and A_2 being environment-dependent constants.

Assume that all UAVs are assigned the same bandwidth B, across different frequency bands, and each UAV can collect data from multiple SNs in a cell concurrently by employing Frequency Division Multiple Access. Let $b^{in}(t)$ be the bandwidth that UAV n allocates to SN i at time t. Let $C^{in}(t)$ be a binary indicator, where $C^{in}(t) = 1$ indicates that UAV n is hovering above the cell containing SN i to collect data at time t, and $C^{in}(t) = 0$ indicates the otherwise. We have the following constraints for $b^{in}(t)$

$$\sum_{i\in\mathcal{I}} b^{in}(t) \leq B \text{ and } b^{in}(t) \leq C^{in}(t)B, \forall n \in \mathcal{N}, t \in \mathcal{T}^n. \quad (6)$$

The instantaneous achievable upload data rate (bps) from SN i to UAV n at time t can be calculated as

$$R^{in}(t) = b^{in}(t)\log_2\left(1 + |h^{in}(t)|^2 P_s/(b^{in}(t)N_0)\right) \quad (7)$$

where P_s is the fixed transmit power of SNs and N_0 is the noise power spectral density.

Now, we consider the time $t = t^n[k]$ when the n-th UAV makes its k-th decision. Assume that the channels remain unchanged during UAV's hovering, i.e., $R^{in}(t) = R^{in}(t^n[k])$ for $t^n[k] \leq t \leq t^n[k+1]$. The hovering time above $\mathbf{w}^n[k]$ can then be determined as

$$\tau^n_{\mathrm{H}}[k] = \max_{i\in\mathcal{I}}\left\{\frac{C^{in}(t^n[k])D^i}{R^{in}(t^n[k])} \;\middle|\; R^{in}(t^n[k]) > 0\right\}. \quad (8)$$

Let $b^{in}[k] = b^{in}(t^n[k])$ and $\mathbf{b}^n = \{b^{in}[k] : i \in \mathcal{I}, k \in \mathcal{K}^n\}$ be the bandwidth allocation strategy of UAV n. Given trajectories $\mathbf{w}$ and bandwidth allocation strategies $\mathbf{b} = [\mathbf{b}^1, \mathbf{b}^2, \ldots, \mathbf{b}^N]$, we have the total collected data given by

$$\Phi(\mathbf{w},\mathbf{b}) = \sum_{n\in\mathcal{N}}\sum_{k\in\mathcal{K}^n}\sum_{i\in\mathcal{I}} C^{in}(t^n[k])D^i. \quad (9)$$

Finally, we have a constraint representing termination conditions of UAVs as follows

$$(E^n[K^n+1]-\epsilon)\left(\sum\nolimits_{i\in\mathcal{I}} D^i - \Phi(\mathbf{w},\mathbf{b})\right) \leq 0, \forall n \in \mathcal{N} \quad (10)$$

where $E^n[K^n+1]$ is the remaining energy of the n-th UAV when arriving at its final destination $\mathbf{w}^n_{\mathrm{F}}$. This constraint forces UAVs to continue exploring and collecting data until either their energy falls below the safe level or all data is collected.

C. Problem formulation

Our aim is to jointly optimize the trajectories $\mathbf{w}$ and bandwidth allocation strategies $\mathbf{b}$ of all N UAVs, such that the overall energy efficiency is maximized. This problem can be mathematically formulated as follows

$$\begin{aligned} \max_{\mathbf{w},\mathbf{b}} \quad & \frac{\Phi(\mathbf{w},\mathbf{b})}{\psi(\mathbf{w},\mathbf{b})} \\ \text{s.t.} \quad & (1),(2),(3),(6),(10) \end{aligned} \quad (\mathrm{P})$$

where $\Phi(\mathbf{w},\mathbf{b})$ and $\psi(\mathbf{w},\mathbf{b})$ are given in (9) and (4), respectively. Problem (P) is intractable due to $i)$ high level of uncertainty of CSI and stochasticity of data collection demands; $ii)$ non-deterministicity of the planning horizon K^n and non-convexity in the objective function and constraints; $iii)$ uncertainty in such multi-agent systems, where the operation of each UAV influences the environment and consequently affects the decisions of other UAVs. Besides, effective cooperation among UAVs is hindered by limitations in their observation and communication abilities. Our aim in this study is to provide a sub-optimal solution to (P) by decomposing this problem into two sub-problems, including cooperative trajectory optimization and bandwidth allocation. While the latter can be solved based on local observation at each hovering point using similar approach in [12], planning UAVs' trajectories poses more complexity, especially when decisions are made asynchronously among UAVs.

III. Trajectory Optimization under Dec-POSMDP

A. Dec-POSMDP framework for trajectory optimization

The trajectory optimization problem can be represented by a tuple $\langle\mathcal{N}, \mathcal{S}, \mathcal{A}, P, R, \mathcal{Z}, \gamma\rangle$, where $\mathcal{S}$ is the global state space of the environment, $\mathcal{A} = \times_{n\in\mathcal{N}}\mathcal{A}^n$ is the joint action space, $P : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to [0,1]$ is the environment transition kernel, $R : \mathcal{A}\times\mathcal{S} \to \mathbb{R}$ is a shared reward function contributed by all UAVs, $\mathcal{Z} = \times_{n\in\mathcal{N}}\mathcal{Z}^n$ is the joint observation space, and finally, γ is the discount factor. Let $z^n[k] \in \mathcal{Z}^n$ be the local observation received by the n-th UAV and $u^n[k] \in \mathcal{A}^n$ be the action chosen by this UAV at time $t^n[k]$. We have the joint action-observation history of the n-th UAV until $t^n[k]$ as follows

$$H^n[k] = (z^n[1], u^n[1], z^n[2], \ldots, u^n[k-1], z^n[k]). \quad (11)$$

Let $\mathcal{T} = \bigcup_{n\in\mathcal{N}}\mathcal{T}^n = \{\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_m, \ldots\}$ be the set of all decision-making instants of all UAVs sorted in the non-decreasing order. Let s_m denote the global state captured at time $\hat{t}_m$ and $\mathbf{u}_m = \left(u^1_m, u^2_m, \ldots, u^N_m\right)$ be the joint action of all UAVs taken at this time. The joint action $\mathbf{u}_m$ here includes two parts, including a new action calculated by a UAV that finishes its action at $\hat{t}_m$ and the on going actions of other UAVs calculated before $\hat{t}_m$. Let define the reward function for executing the joint action $\mathbf{u}_m$ in state s_m at time $\hat{t}_m$ by

$$R(s_m, \mathbf{u}_m) = \int_{\hat{t}_m}^{\hat{t}_{m+1}} \gamma^{t-\hat{t}_m}\widehat{R}(s_m, \mathbf{u}_m, t)dt \quad (12)$$

where $\hat{R}(s_m, \mathbf{u}_m, t)$ is the total instantaneous reward received by all UAVs at time t for taking action $\mathbf{u}_m$ in state s_m. Precisely, $R(s_m, \mathbf{u}_m)$ is the total reward accumulated by all UAVs from $\hat{t}_m$ to $\hat{t}_{m+1}$. Let $\pi = \times_{n \in \mathcal{N}} \pi^n$ be the decentralized joint policy where π^n is the local policy of UAV n that maps the local action-observation history $H^n[k]$ to the next action $u^n[k]$. Under the policy π, let define the joint state value function as $V^{\pi}_{\mathtt{tot}}(s_m) = \mathbb{E}_{\pi}\left[\sum_{m=0}^{\infty} \gamma^{\hat{t}_m} R(s_m, \mathbf{u}_m) \mid s_0 = s_m\right]$, and the state-action value function as $Q^{\pi}_{\mathtt{tot}}(s_m, \mathbf{u}_m) = \mathbb{E}_{\pi}\left[\sum_{m=0}^{\infty} \gamma^{\hat{t}_m} R(s_m, \mathbf{u}_m) \mid s_0 = s_m, \mathbf{u}_0 = \mathbf{u}_m\right]$. The problem can then be defined as finding the joint policy π^* to maximize the values of all states, $\pi^* = \underset{\pi}{\text{argmax}} V^{\pi}_{\mathtt{tot}}(s_m), \forall s_m \in \mathcal{S}$.

B. Detailed components

We now transform the trajectory optimization problem into the Dec-POSMDP by defining its components as follows.

1) UAVs' termination conditions: as defined in (10), the termination of a UAV depends on its energy level and the completion of collecting task. For the first condition, each time the UAV calculates its action, it checks if the remaining energy approaches the red lines defined in constraints (2) and (3). If so, the UAV flies directly to its final destination and terminates its mission. To handle the second condition, each UAV n maintains a completion map $G^n \in \{0, 1\}^{H^2 \times 1}$ indicating its belief about the status of each cell. This map is initialized to zeros at the begining, indicating the 'yet-completed' status. Over time, the map is updated after every UAV action based on its observations. Specifically, G^{cn} is set to 1, or 'completed', if the n-th UAV collects all data in the c-th cell or observes that there is no data in the cell. The task is then considered completed by UAV n if all elements of G^n are equal to one.

2) State space: each state $s_m \in \mathcal{S}$ includes positions, energy levels, and completion maps of all UAVs, and total data collection demand remaining at each cell.

3) Observation: the local observation $z^n[k]$ of the n-th UAV includes its remaining energy, current position, completion map G^n, and total data collection demand of each cell within its current observable region.

4) Action space: the action space of the UAV n consists of five actions $\mathcal{A}^n = \{0, 1, 2, 3, 4\}$, which represent the action of hovering to collect data, moving forward, to the right, backward, and to the left, respectively.

5) Communication message between UAVs: When two UAVs are within their communication range, they share positions and synchronize completion maps to assist each other in verifying mission completion.

6) Reward function: as defined in (12), the reward function $R(s_m, \mathbf{u}_m)$ under the joint policy π is the accumulation of the instantaneous reward $\widehat{R}(s_m, \mathbf{u}_m, t)$, contributed by all UAVs. Here, we define the instantaneous reward $\widehat{R}(s_m, \mathbf{u}_m, t)$ as the total data collected by all UAVs at time t and scaled energy efficiency at termination as follows

$$\widehat{R}(s_m, \mathbf{u}_m, t) = \sum_{n \in \mathcal{N}} \sum_{i \in \mathcal{I}} R^{in}(t) + \alpha \Gamma(\pi) \mathbb{1}(s_m, \mathbf{u}_m, t) \quad (13)$$

where α is a scale parameter, $\Gamma(\pi)$ is the energy efficiency achieved under the policy π given in the objective function of (P), and $\mathbb{1}(s_m, \mathbf{u}_m, t)$ is an indicator function indicating that whether taking action $\mathbf{u}_m$ in state s_m lead to the termination of the last UAV at time t. In (13), the first term serves as a shaping function to support the main learning goal in the second term, which is only activated at the last moment when the last UAV arrives to its final destination. Combining (12), (13), and the assumption that channel states remain unchanged during UAV's hovering, we obtain, after some computation, the following analytical form of the reward function

$$R(s_m, \mathbf{u}_m) = \frac{\gamma^{\hat{\tau}_m} - 1}{\ln \gamma} \sum_{\substack{i \in \mathcal{I} \\ n \in \mathcal{N}}} R^{in}\left(\hat{t}_m\right) + \gamma^{\hat{\tau}_m} \alpha \Gamma(\pi) \hat{\mathbb{1}}(s_m, \mathbf{u}_m) \quad (14)$$

where $\hat{\tau}_m = \hat{t}_{m+1} - \hat{t}_m$ is the duration of action $\mathbf{u}_m$ taken in state s_m and $\hat{\mathbb{1}}(s_m, \mathbf{u}_m)$ indicating whether taking action $\mathbf{u}_m$ in state s_m leads to the termination of the last UAV.

IV. Asynchronous Learning Algorithm

The QMIX algorithm aims to learn a centralized action-value function $Q_{\mathtt{tot}}(s, \mathbf{u})$, which is factorized into N individual utility functions $Q^n(H^n, u^n)$ representing the goodness of taking action u^n on history H^n. The principal feature that makes QMIX efficient is the consistent relationship between the deterministic greedy centralized policy and the deterministic decentralized policies resulting from the monotonicity between $Q_{\mathtt{tot}}$ and Q^n, i.e., $\partial Q_{\mathtt{tot}} / \partial Q^n \geq 0, \forall n \in \mathcal{N}$. When such monotonicity is assured, centralized training can be undertaken relying on the following relation

$$\underset{\mathbf{u} \in \mathcal{A}}{\text{argmax}} Q_{\mathtt{tot}}(s, \mathbf{u}) = \left\{ \underset{u^n \in \mathcal{A}^n}{\text{argmax}} Q^n(H^n, u^n) \right\}_{n \in \mathcal{N}}. \quad (15)$$

This result ensures that local actions that improve local Q^n values will have similar effects on the joint action-value function $Q_{\mathtt{tot}}$, enabling decentralized agents to operate independently based on the greedy policy applied to their local Q^n values.

To enable QMIX to be applicable in our asynchronous environment, we make an important modification to this algorithm as follows. Let $\hat{n}$ be the UAV that finishes its action at time $\hat{t}_m$, and H^n_m be the local joint action-observation history of UAV n recorded until $\hat{t}_m$ (i.e., $H^n_m = H^n[k^*]$ where $k^* = \text{argmax}_{k \in \mathcal{K}^n}\{t^n[k] \mid t^n[k] \leq \hat{t}_m\}$). As mentioned in the preceding section, there is a new action $u^{\hat{n}}_m$ calculated by UAV $\hat{n}$ at $\hat{t}_m$, while other UAVs' actions continue to be executed. Let $\mathbf{u}^{-\hat{n}}_m$ denote the set of these ongoing actions. Neglecting the order of UAVs, we have the joint action at time $\hat{t}_m$ given by $\mathbf{u}_m = \{u^{\hat{n}}_m\} \cup \mathbf{u}^{-\hat{n}}_m$. The aim of the asynchronous algorithm is to learn a centralized action-value function $Q_{\mathtt{tot}}(s_m, \mathbf{u}_m | \mathbf{u}^{-\hat{n}}_m)$ conditioned on the ongoing actions $\mathbf{u}^{-\hat{n}}_m$. We have the following result for our asynchronous algorithm.

Lemma 1. *Given that $\frac{\partial Q_{\mathtt{tot}}}{\partial Q^n} \geq 0 \; \forall n \in \mathcal{N}$, and that*

$$\begin{aligned} &Q_{\mathtt{tot}}\left(s_m, \boldsymbol{u}_m | \boldsymbol{u}^{-\hat{n}}_m\right) \\ &\quad = Q_{\mathtt{tot}}\left(Q^{\hat{n}}(H^{\hat{n}}_m, u^{\hat{n}}_m), \{Q^n(H^n_m, u^n_m)\}_{n \in \mathcal{N} \setminus \{\hat{n}\}}\right), \end{aligned}$$

then we have

$$\underset{\boldsymbol{u}_m \in \mathcal{A}}{\text{argmax}} Q_{tot}\left(s_m, \boldsymbol{u}_m | \boldsymbol{u}_m^{-\hat{n}}\right) = \{\underset{u_m^{\hat{n}} \in \mathcal{A}^{\hat{n}}}{\text{argmax}} Q^{\hat{n}}\left(H_m^{\hat{n}}, u_m^{\hat{n}}\right)\} \cup \boldsymbol{u}_m^{-\hat{n}}.$$

The proof to Lemma 1 is excluded due to space limitation. This lemma implies that once the monotonicity between $Q_{\texttt{tot}}$ and Q^n is established, the local deterministic greedy policies remain applicable for agents to calculate their actions in asynchronous environments.

The network architecture in our algorithm is retained as in synchronous QMIX [9]. Specifically, the architecture includes two components: i) agent networks that take local action-observation histories and the agent indices as input and outputs the local Q^n values, and ii) a mixing network that uses the global state information to produce $Q_{\texttt{tot}}$ from N local Q^n values. To establish the monotonic relationship between $Q_{\texttt{tot}}$ and Q^n, all weights of the mixing network are constrained to be non-negative, which is realized by using a hypernetwork. The training is end-to-end based on a replay buffer, aiming to minimize the loss function

$$\mathcal{L}(\theta_m) = \mathbb{E}\left[\left(y_{\texttt{tot}} - Q_{\texttt{tot}}(s_m, \mathbf{u}_m | \mathbf{u}_m^{-\hat{n}}; \theta_m)\right)^2\right] \tag{16}$$

where $y_{\texttt{tot}}$ is the target value estimated by one-step bootstrapping as

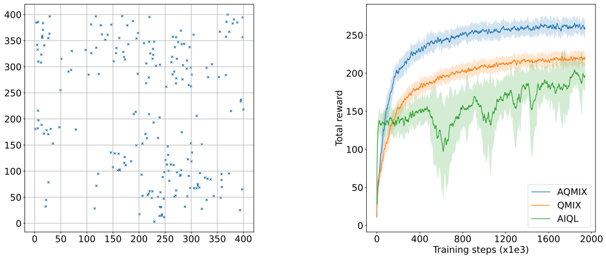
$$y_{\texttt{tot}} = R(s_m, \mathbf{u}_m) + \gamma^{\hat{\tau}_m} \max_{\mathbf{u}_{m+1}} Q_{\texttt{tot}}(s_{m+1}, \mathbf{u}_{m+1} | \mathbf{u}_{m+1}^{-\hat{n}}; \theta_m^-)$$

wherein θ_m and θ_m^- is the parameters of the primary and the target networks. It's worth noting that, although the network architecture is similar as in synchronous QMIX, the sampling process of our algorithm is entirely different, i.e., at every transition step, only the observation of the agent that finishes its action is updated, while those of the other agents remain unchanged.

V. Simulation Results

A. Simulation description

The proposed method is evaluated on a monitored area of 400×400m, which is divided into 8×8 cells of 50×50m each. The distribution of SNs over the collecting area is illustrated in Fig. 2a. To generate data collection demands, we first generate the number of cells containing data following Poisson distribution with the mean value $\lambda = 0.3H^2$. Once the number of cells containing data is determined, these cells are placed randomly in the monitored area. For all SNs located in the selected cells, the data size is generated uniformly in $[0.1, 1.0]$ kbits. We set the transmit power of SNs $P = 0.1$W, attenuation due to NLoS $\beta = 0.2$, total bandwidth $B = 1$MHz, noise power spectral density $N_0 = -110$dBm, and the pathloss exponent $\eta = 2.6$. We use two UAVs operating at altitude $h = 100$m with different velocities of 10m/s and 20m/s. Initial locations and final destinations of all UAVs are set at the same position, at the center of the bottom-left cell depicted in Fig. 2a. The observable region of each UAV is an area of 3×3 cells centered at its location. In all experiments, UAVs are trained with $E_{\max}^n = 500$kJ and inter-UAV communication range of 200m. All other UAVs' parameters are retained as in [10].



(a)

(b)

Fig. 2: Distribution of SNs (a) and learning curves of algorithms averaged across 10 runs with different random initializations (b).

The proposed method is named as Asynchronous-QMIX (AQMIX). To evaluate the performance of this method, the forllowing baselines are used:

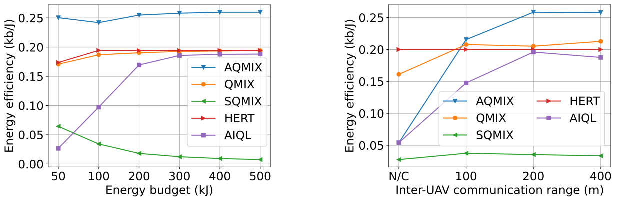
- Independent learning (AIQL): this is a fully decentralized baseline obtained by removing the mixing network from the architecture of AQMIX.
- QMIX [6]: this algorithm was designed only for synchronous environments. To be applicable in our asynchronous setting, we assume that there is a synchronizer between UAVs, i.e., a UAV has to wait if it completes its action before the others, and then all UAVs calculate their new actions at the same time. However, since such a synchronizer is not available in practice, the solely aim of this baseline is to demonstrate the convergence behavior of AQMIX.
- Semi-QMIX (SQMIX): this baseline involves deploying synchronous policies learned by QMIX [9] in an asynchronous environment, i.e., allowing UAVs to make their decisions without waiting for others.
- Heuristic (HERT): a very naive but feasible solution to (P) is to partition the area into multiple sectors and assign each part to one UAV. UAVs then fly over their assigned sub-areas, exploring and collecting data cell by cell.

All learning methods are trained with the same computational budget of 2e6 steps per UAV. We use learning rate of 1e-4, discount factor of $\gamma = 0.99$, batch size of 64, replay buffer size of 5e4, and target network update rate of 1e-2. Each hidden layer of the policy networks includes 128 neurons, while this number in the mixing network is 256. We adopt the equal bandwidth allocation during training to mitigate training time. After training, bandwidth allocation is optimized for testing using a similar method as described in [12].

B. Performance comparison

To evaluate the convergence behavior of algorithms, Fig. 2b plots the total reward during training of the three learning methods, where each line is the average result of ten different runs, and shaded areas represent the standard deviation. The results indicate that AQMIX and QMIX clearly outperform

AIQL. The fluctuations in the performance of AIQL suggest that it has failed to exploit the benefits from inter-UAV communication in an organized manner. Regarding two remaining methods, AQMIX achieves a higher reward compared to QMIX. This is because QMIX requires UAVs to wait for synchronization with others before making decisions, leading to inefficient use of hovering energy. More importantly, Fig. 2b also indicates that our modification to QMIX has successfully enabled this algorithm to learn in asynchronous environments, demonstrated by the stable and highly competitive learning performance of AQMIX.



(a) Varying energy, fixed com. range

(b) Fixed energy, varying com. range

Fig. 3: Efficiency achieved by algorithms, averaged over 100 testing scenarios. 'N/C' in (b) denotes non-communication.

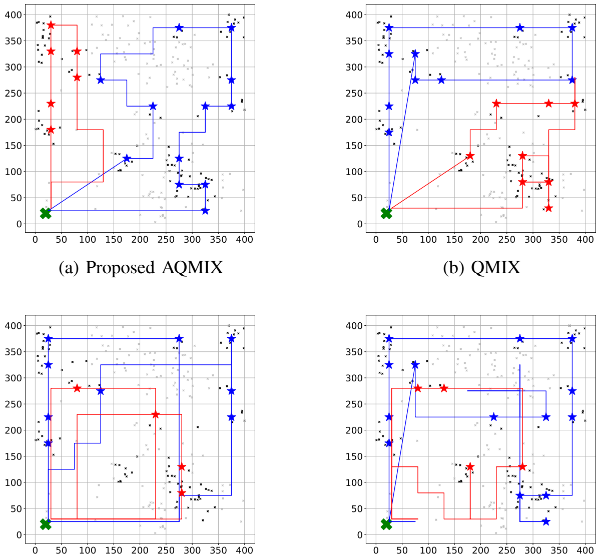
Fig. 3a shows the testing results of best policies obtained by all methods on the same set of 100 collection demand scenarios, with $E^n_{\max}$ ranging from 50kJ to 500kJ. At all energy levels, AQMIX algorithm demonstrates outstanding performance compared to all remaining baselines. Fig. 3b illustrates testing results of policies trained at different inter-UAV communication ranges. As the communication range expands, the effectiveness of methods tends to improve. The performance of AQMIX and QMIX initially falls short of the heuristic and then surpassing this baseline. In both experiments above, AIQL and SQMIX shows worst performance. Fig. 4 visualizes trajectories generated by all learning-based methods for one of the testing scenarios. This figure once again indicates the superiority of the proposed method AQMIX, as demonstrated by the cooperation between UAVs, thereby avoiding trajectory overlaps as observed in other algorithms.

VI. Conclusion

In this study, we have proposed a comprehensive solution to the cooperative UAVs data collection problem in remote areas. A central focus has been the intricate challenges of incomplete information and asynchronous decision-making among UAVs. To address these challenges, we have introduced an asynchronous multi-agent learning framework based on centralized training, which has outperformed existing multi-agent based solutions. Beyond the scope of the investigated case study, we expect that our novel modeling methodology and proposed solution could prove helpful in analogous systems where asynchrony exists.

Acknowledgement

This work was funded by the Luxembourg National Research Fund (FNR), grant reference FNR/C22/IS/17220888/RUTINE.



(a) Proposed AQMIX

(b) QMIX

(c) SQMIX

(d) AIQL

Fig. 4: Trajectories generated by learned policies. Green 'X' symbols represent UAV's initial/final locations, stars represent hovering locations, small black and gray dots represents SNs containing and non-containing data, respectively.

References

[1] C. You and R. Zhang, "3d trajectory optimization in rician fading for uav-enabled data harvesting," *IEEE Transactions on Wireless Communications*, vol. 18, no. 6, pp. 3192–3207, 2019.

[2] C. Zhan and Y. Zeng, "Completion time minimization for multi-uav-enabled data collection," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4859–4872, 2019.

[3] Y. Wang, Z. Gao, J. Zhang, X. Cao, D. Zheng, Y. Gao, D. W. K. Ng, and M. Di Renzo, "Trajectory design for uav-based internet of things data collection: A deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3899–3912, 2021.

[4] J. Hu, H. Zhang, L. Song, R. Schober, and H. V. Poor, "Cooperative internet of uavs: Distributed trajectory design by multi-agent deep reinforcement learning," *IEEE Transactions on Communications*, vol. 68, no. 11, pp. 6807–6821, 2020.

[5] G. Chen, X. B. Zhai, and C. Li, "Joint optimization of trajectory and user association via reinforcement learning for uav-aided data collection in wireless networks," *IEEE Transactions on Wireless Communications*, 2022.

[6] X. Wang, M. Yi, J. Liu, Y. Zhang, M. Wang, and B. Bai, "Cooperative data collection with multiple uavs for information freshness in the internet of things," *IEEE Transactions on Communications*, 2023.

[7] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of markov decision processes," *Mathematics of operations research*, vol. 27, no. 4, pp. 819–840, 2002.

[8] S. Omidshafiei, A.-A. Agha-Mohammadi, C. Amato, and J. P. How, "Decentralized control of partially observable markov decision processes using belief space macro-actions," in *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 5962–5969, IEEE, 2015.

[9] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *Journal of Machine Learning Research*, vol. 21, no. 178, pp. 1–51, 2020.

[10] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing uav," *IEEE transactions on wireless communications*, vol. 18, no. 4, pp. 2329–2345, 2019.

[11] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal lap altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[12] T. X. Vu, S. Chatzinotas, and B. Ottersten, "Dynamic bandwidth allocation and precoding design for highly-loaded multiuser miso in beyond 5g networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 3, pp. 1794–1805, 2022.