SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2024, Vol. 67, Iss. 3, 132302:1–132302:21
https://doi.org/10.1007/s11432-023-3906-3

Joint UAV trajectory and communication design with heterogeneous multi-agent reinforcement learning

Xuanhan ZHOU[1], Jun XIONG[1], Haitao ZHAO[1*], Xiaoran LIU[1], Baoquan REN[2], Xiaochen ZHANG[1], Jibo WEI[1] & Hao YIN[2]

[1]*College of Electronic Science and Technology, National University of Defense Technology, Changsha* 410073*, China;*
[2]*Systems Engineering Institute, Academy of Military Sciences PLA, Beijing* 100091*, China*

Received 16 May 2023/Revised 28 August 2023/Accepted 30 November 2023/Published online 20 February 2024

Abstract Unmanned aerial vehicles (UAVs) are recognized as effective means for delivering emergency communication services when terrestrial infrastructures are unavailable. This paper investigates a multi-UAV-assisted communication system, where we jointly optimize UAVs' trajectories, user association, and ground users (GUs)' transmit power to maximize a defined fairness-weighted throughput metric. Owing to the dynamic nature of UAVs, this problem has to be solved in real time. However, the problem's non-convex and combinatorial attributes pose challenges for conventional optimization-based algorithms, particularly in scenarios without central controllers. To address this issue, we propose a multi-agent deep reinforcement learning (MADRL) approach to provide distributed and online solutions. In contrast to previous MADRL-based methods considering only UAV agents, we model UAVs and GUs as heterogeneous agents sharing a common objective. Specifically, UAVs are tasked with optimizing their trajectories, while GUs are responsible for selecting a UAV for association and determining a transmit power level. To learn policies for these heterogeneous agents, we design a heterogeneous coordinated QMIX (HC-QMIX) algorithm to train local Q-networks in a centralized manner. With these well-trained local Q-networks, UAVs and GUs can make individual decisions based on their local observations. Extensive simulation results demonstrate that the proposed algorithm outperforms state-of-the-art benchmarks in terms of total throughput and system fairness.

Keywords unmanned aerial vehicle (UAV), trajectory design, resource allocation, multi-agent deep reinforcement learning (MADRL), heterogeneous agents

1 Introduction

In recent years, unmanned aerial vehicles (UAVs) have emerged as a promising technology to supplement existing cellular systems or provide emergency communication services [1,2]. When used as base stations (BSs), UAVs are more likely to establish line-of-sight (LoS) links with ground users (GUs) [3]. Moreover, they can be easily deployed and dynamically adjust their locations to improve channel quality, making them an attractive option for reliable wireless transmission [4,5]. To overcome the size, weight, and power limitations of individual UAVs, deploying multiple UAVs as mobile BSs is a common strategy to extend network coverage and enhance overall communication capability [6].

Despite the advantages mentioned above, optimizing a multi-UAV-assisted communication system presents greater challenges than its terrestrial counterpart [7]. One crucial issue involves the mitigation of substantial cross-link interference stemming from LoS-dominant air-to-ground channels [8]. The high maneuverability of UAV-BSs introduces a new degree of freedom for interference management. However, when addressing the joint optimization of UAV trajectory and communication, the problem complexity escalates. Furthermore, the fluctuating network topology and dynamic channel conditions caused by UAV movements necessitate frequent updates of control signals. Although conventional optimization-based algorithms can achieve optimal or close-to-optimal solutions for these problems [9–21], their applicability in real-time scenarios is hindered by their computational complexity. Moreover, these approaches rely

* Corresponding author (email: haitaozhao@nudt.edu.cn)

on centralized control and global information, which is usually infeasible in multi-UAV systems without central controllers.

Multi-agent deep reinforcement learning (MADRL) [22] presents a potential solution to the above problems. Deep reinforcement learning (DRL) [23] learns optimal policies from experience and rewards collected through interaction with environments. After sufficient training, online decision-making based on deep neural networks (DNNs) is significantly faster than optimization-based algorithms [24]. DRL-based methods for multi-UAV communications [25, 26] typically learn a centralized control policy for the entire system by regarding all UAVs as a single agent. These methods cannot be deployed in a distributed manner and suffer from the curse of dimensionality as the number of UAVs increases. In contrast, MADRL enables distributed implementation by treating each UAV as an individual decision-maker, making it well-suited for multi-UAV systems [27, 28]. Although numerous studies have explored MADRL applications in multi-UAV communications [29–34], previous research focuses solely on UAV agents, overlooking the potential performance improvements achieved by treating both UAVs and GUs as agents.

The physical heterogeneity of UAV and GU agents necessitates learning algorithms capable of addressing the discrepancies in observation and action spaces. However, most existing MADRL algorithms [35–39] have been tailored for homogeneous multi-agent scenarios, relying on shared policy parameters to streamline the training process. When dealing with heterogeneous agents, these algorithms prove insufficient to effectively tackle the increased challenges posed. Firstly, the multi-agent credit assignment issue is particularly complex in heterogeneous multi-agent scenarios, where agents possess diverse capabilities, roles, and behaviors [40]. The need to accurately distribute credit becomes more pronounced as agents' disparate abilities and roles can lead to imbalances in contribution, potentially affecting the overall system performance. Additionally, the interdependence among heterogeneous agents leads to more complex dynamics, where an action by one type of agent might have cascading effects on the other type [41]. In such cases, identifying optimal policies that leverage the strengths of both agent types while mitigating their weaknesses requires further investigation.

In this paper, we propose a heterogeneous MADRL algorithm capable of learning policies for heterogeneous agents with a shared reward. We apply it to address the interference management problem in multi-UAV-assisted communications. Our contributions are summarized as follows:

(1) We formulate an air-ground interference coordination problem, where UAVs' trajectories, user association, and GUs' power control are jointly optimized. To achieve a balance between the overall system throughput and fairness among GUs, we define a fairness-weighted throughput metric as the optimization objective.

(2) Given the real-time and decentralized requirement, we model UAVs and GUs as individual agents and reformulate the problem as a decentralized partially observable Markov decision process (Dec-POMDP). Specifically, UAV agents are tasked with optimizing their trajectories, whereas GU agents are responsible for selecting the optimal UAV for association and determining an appropriate transmit power level.

(3) To learn the policies of agents, we propose a novel heterogeneous coordinated QMIX (HC-QMIX) algorithm. A shared mixing network is constructed to train a centralized Q-function that can be factored into individual Q-functions for both UAV and GU agents. Equipped with well-trained local Q-functions, UAVs and GUs are enabled to make individual decisions. Moreover, this approach effectively mitigates the multi-agent credit assignment issue.

The rest of this paper is organized as follows. Section 2 reviews existing research on UAV-assisted communications. Section 3 describes the system model and formulated optimization problem. Section 4 reformulates the optimization problem into a Dec-POMDP. Section 5 presents details of the HC-QMIX algorithm. Section 6 compares the proposed algorithm with benchmarks via simulation. Finally, Section 7 concludes this work.

2 Related work

Early research on UAV-assisted wireless communications primarily focuses on optimizing UAV trajectory and resource allocation using optimization-based algorithms [9–17]. In [9, 10], the UAV trajectory optimization problem is investigated for a mobile relaying system and an energy-efficient communication system with a single UAV and a single GU. These studies formulate a non-convex problem and

employ the successive convex approximation (SCA) technique [42] to obtain locally optimal solutions. For multi-user scenarios, Wu et al. [11] studied an orthogonal frequency-division multiple access network, using block coordinate descent (BCD) [43] and SCA techniques to update UAV trajectory and resource allocation variables. Thereafter, BCD and SCA have been applied for joint UAV trajectory and communication optimization in various contexts, such as UAV-aided data collection [12, 18] and UAV-assisted mobile edge computing (MEC) [13, 14]. Several studies investigate the more complex multi-UAV communications [15–17]. For example, Mozaffari et al. [15] considered an energy-efficient data collection system, where the mobility of UAVs, user association, and uplink transmit power are jointly optimized to minimize the total energy consumption of ground devices. Furthermore, the joint UAV trajectory design, user scheduling, and power control problem are studied in [16], aiming to maximize the minimum throughput over all GUs. Although the aforementioned studies provide various solutions for the non-convex UAV trajectory and communication optimization problem, these algorithms suffer from high complexity and do not meet the real-time and decentralized requirement of multi-UAV communications.

MADRL techniques are applicable to problems where online solutions are required, particularly in the absence of a central controller. Recently, there emerge a large number of studies applying DRL/MADRL for multi-UAV assisted communications [25, 26, 29–34, 44]. In [25], DRL is used to optimize UAV trajectories, with a DNN-based agent centrally controlling the entire multi-UAV system. Further, Zhang et al. [26] developed a DRL-based self-regulation approach that accounts for changes in UAV lineups. These studies treat all UAVs as a single agent and utilize a centralized movement control policy, making distributed implementation difficult. Moreover, centralized learning methods have poor scalability, as the state space in multi-UAV systems grows exponentially with the number of UAVs. Alternatively, Zhang et al. [29] considered each UAV as an independent learner, training a trajectory control policy for each UAV by treating other agents as part of the environment. This method enables distributed deployment for trajectory optimization of individual UAVs. Similar approaches are adopted in [30, 31] for the multi-UAV resource allocation problem. Given the predefined UAV trajectory, Cui et al. [30] focused on throughput maximization through user association and power control, whereas Yuan et al. [31] minimized the total energy consumption via user scheduling. To further enhance the spectrum efficiency, Zhong et al. [32] employed the non-orthogonal multiple access (NOMA) at UAVs and optimized the trajectory and power allocation of each UAV independently based on deep Q-network (DQN) algorithm [23]. Although these independent learning approaches [29–32] offer advantages in terms of scalability and distributed implementation, they fail to consider the mutual influence among agents and may not converge. To address this issue, Zhang et al. [33] incorporated decision information exchange among UAVs using graph neural networks (GNNs). In [34, 44], the centralized training and distributed execution framework is employed to promote cooperation among UAVs and improve converged performance. Qin et al. [34] adopted the multi-agent deep deterministic policy gradient (MADDPG) algorithm [22] to optimize UAV trajectory, where each UAV learns a centralized critic based on the observations and actions of all agents. Then, Ding et al. [44] further proposed a joint UAV trajectory design and user access control optimization algorithm based on MADDPG, by treating UAVs and GUs as agents with different objectives.

Most existing studies [25, 26, 29–34] primarily focus on trajectory design or resource allocation of UAV agents. However, the interference management in uplink transmission not only depends on UAV-BSs, but also relies on GUs. While Ding et al. [44] considered both UAV trajectory and GUs' access control, their adoption of a non-orthogonal scheme for spectrum allocation precludes the investigation of interference management issues. Moreover, fully cooperative settings, where all agents share the same objective, present a multi-agent credit assignment challenge [36]. In heterogeneous multi-agent scenarios, this issue is exacerbated due to the inherent variation in agents' skill levels and their distinct functions within the system. MADDPG algorithm proves inadequate for addressing this challenge, potentially resulting in degraded performance under partially observable conditions. While a limited number of investigations have explored heterogeneous MADRL in domains like multi-robot systems [40, 41], traffic lights control [45], and taxation policy design [46], no prior work has modeled both UAVs and GUs as agents for cooperative interference management in multi-UAV assisted communications. To fully exploit the intricate interference relationship between UAVs and GUs, there is an urgent demand for a novel learning framework applicable to heterogeneous multi-agent settings, which motivates our work.

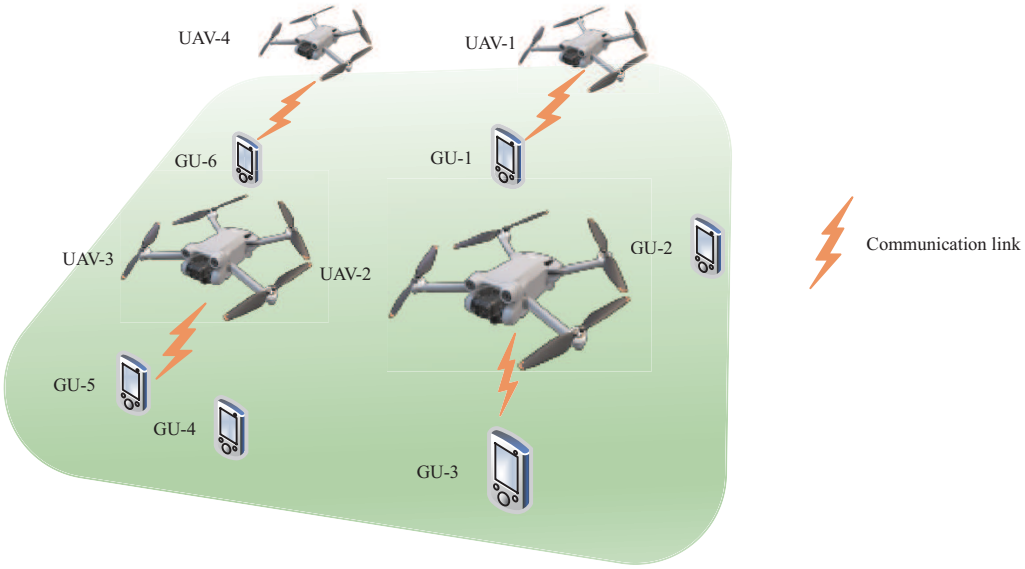

Figure 1 (Color online) Illustrative example of the system model. Due to the unavailability of the ground base station (GBS), UAVs are employed as mobile BSs to provide service to GUs. There are 4 UAVs and 6 GUs. Specifically, GU-1, GU-3, GU-5, and GU-6 establish communication with UAVs, whereas GU-2 and GU-4 are currently not in communication service.

3 System model and problem formulation

3.1 System model

As illustrated in Figure 1, we consider a multi-UAV-assisted uplink communication system. In this system, the ground base station (GBS) is unavailable. To confront this situation, we deploy multiple UAVs as mobile BSs to cooperatively provide communication services for GUs. The entire service duration is evenly divided into time slots of length τ. Within each time slot, every GU selects an appropriate UAV for the association. Subsequently, UAVs allocate dedicated frequency resources for their associated GUs. Once a GU is scheduled by a UAV, it modulates its data into a radio signal and transmits the signal to the UAV at a specific power level. The transmission quality is influenced by the UAV's position and the GU's transmit power. To improve system communication performance, our objective is to optimize the trajectory for each UAV, determine the most suitable UAV for each GU to establish an association, and identify their optimal transmit power. Prior to introducing our proposed approach, we provide a comprehensive depiction of the system model in the following. Main notations defined in this section are shown in Table 1.

3.1.1 *Mobility model*

Let $\mathcal{U} = \{1, \ldots, U\}$ and $\mathcal{M} = \{1, \ldots, M\}$ denote the sets of UAVs and GUs, respectively, and $\mathcal{T} = \{1, \ldots, T\}$ denote the set of time slots. The position of each GU-m remains fixed at $\boldsymbol{w}^m = [x^{\text{user},m}, y^{\text{user},m}, 0]$. At time slot t, the position of UAV-u is given by $\boldsymbol{q}_t^u = [x_t^{\text{uav},u}, y_t^{\text{uav},u}, z_t^{\text{uav},u}]$, where $z_t^{\text{uav},u} = H^{\text{uav}}$ represents the constant flying altitude. Thus, the UAV trajectory can be described by a T-length sequence $\{\boldsymbol{q}_t^u\}_{t=1}^T$. Besides, the UAV velocity can be approximated by $\boldsymbol{v}_t^u = (\boldsymbol{q}_t^u - \boldsymbol{q}_{t-1}^u)/\tau$. We assume a constant UAV speed V, i.e,

$$||\boldsymbol{q}_t^u - \boldsymbol{q}_{t-1}^u||/\tau = V, \ \forall\ u \in \mathcal{U}, \ \forall\ t \neq 1 \in \mathcal{T}. \tag{1}$$

In addition, trajectories of any two UAVs must adhere to collision avoidance constraints, expressed by

$$||\boldsymbol{q}_t^u - \boldsymbol{q}_t^{u'}|| \geqslant l_{\min}, \ \forall u \neq u' \in \mathcal{U}, \ \forall t \in \mathcal{T}, \tag{2}$$

where $l_{\min}$ denotes the minimum distance between UAVs.

3.1.2 *Channel model*

Previous studies have demonstrated that air-to-ground channels are dominated by LoS links in many practical scenarios [3]. Therefore, we employ the free-space path loss model to calculate the channel

Table 1 Notations in system model

Symbol	Definition
U, M	Numbers of UAVs/GUs
$\mathcal{U}$, $\mathcal{M}$	Sets of UAVs/GUs
τ	Duration of each time slot (s)
T	Number of time slots
$\mathcal{T}$	Sets of time slots
$\boldsymbol{q}_t^u$, $\boldsymbol{w}^m$	Position of UAV-u/GU-m (m)
$l_t^{u,m}$	Distance between UAV-u and GU-m (m)
$\boldsymbol{g}_t^{u,m}$	Channel gain from GU-m to UAV-u
B	Bandwidth of available spectrum (Hz)
$\alpha_t^{u,m}$	Association relation between UAV-u and GU-m
$\beta_t^{u,m}$	Scheduling relation between UAV-u and GU-m
p_t^m	Transmit power of GU-m
$P_{\max}$	Maximum transmit power of GUs
N_0	Power spectral density of additive white Gaussian noise
I_t^m	Interference received by GU-m
$\mathrm{SINR}_t^{u,m}$	SINR received by UAV-u from GU-m during time slot t
γ_t^m	Transmission rate of GU-m at time slot t
Γ_t^m	Total throughput of GU-m over t time slots
Γ_t	Total throughput of the entire system over t time slots
$\bar{\Gamma}_t^m$	Average throughput achieved by GU-m over previous t time slots
η_t	Fairness index at time slot t
Γ_f	Fairness-weighted throughput over the given period
ξ	Type of agent (UAV or GU)
$\mathcal{U}_t^{\xi,i}$, $\mathcal{M}_t^{\xi,i}$	Sets for neighboring UAVs or GUs of agent (ξ, i)
$R_{\mathrm{sense}}^{\mathrm{uav}}/R_{\mathrm{sense}}^{\mathrm{user}}$	Sensing range of UAVs/GUs

power gain. This model accounts for the direct signal propagation in the absence of obstacles or other impairments. The channel gain between a UAV and a GU is determined by the distance between them, following the inverse square law. The channel gain between UAV-u and GU-m is computed by

$$\boldsymbol{g}_t^{u,m} = \frac{\rho_0}{(l_t^{u,m})^2}, \tag{3}$$

where $l_t^{u,m} = ||\boldsymbol{q}_t^u - \boldsymbol{w}^m||$ denotes the distance between UAV-u and GU-m, and ρ_0 represents the channel power gain at the reference distance of 1 m.

3.1.3 *Communication model*

To ensure seamless communication coverage even in highly dynamic scenarios, all UAVs and GUs are equipped with omni-directional antennas [6]. For user association, a binary variable $\alpha_t^{u,m}$ is introduced, where $\alpha_t^{u,m} = 1$ indicates GU-m associates with UAV-u at time slot t and $\alpha_t^{u,m} = 0$ otherwise. Each GU can associate with a maximum of one UAV per time slot, as constrained by

$$\sum_{u=1}^{U} \alpha_t^{u,m} \leqslant 1, \ \forall m \in \mathcal{M}, \ \forall t \in \mathcal{T}. \tag{4}$$

We adopt a heuristic scheduling scheme. If multiple GUs associate with the same UAV, the UAV prioritizes scheduling the GU with the strongest channel quality to provide communication service. This procedure can be represented by a binary indicator $\beta_t^{u,m} \in \{0, 1\}$ that shows whether GU-m is scheduled by UAV-u at time slot t.

GUs' transmit power levels are adjustable within defined bounds. Let p_t^m denote the uplink transmit power of GU-m at time slot t. p_t^m is subject to the following constraint:

$$0 \leqslant p_t^m \leqslant P_{\max}, \ \forall m \in \mathcal{M}, \ \forall t \in \mathcal{T}, \tag{5}$$

where $P_{\max}$ denotes the maximum transmit power. The signal-to-interference-plus-noise ratio (SINR) received by UAV-u from GU-m during time slot t is given by

$$\mathrm{SINR}_t^{u,m} = \frac{\alpha_t^{u,m}\beta_t^{u,m}p_t^m \boldsymbol{g}_t^{u,m}}{N_0 B + I_t^m}, \tag{6}$$

where I_t^m is computed as

$$I_t^m = \sum_{m'=1,m'\neq m}^{M}\sum_{u'=1,u'\neq u}^{U} \alpha_t^{u',m'}\beta_t^{u',m'}p_t^{m'}\boldsymbol{g}_t^{u,m'},$$

representing interference from all other GUs. N_0 denotes the power spectral density (PSD) of additive white Gaussian noise (AWGN), and B is the spectrum bandwidth. In accordance with Shannon's theorem, the achievable uplink transmission rate of GU-m at time slot t is computed by

$$\gamma_t^m = \sum_{u=1}^{U} B\log\left(1+\mathrm{SINR}_t^{u,m}\right). \tag{7}$$

As a result, the total achievable throughput of GU-m and the entire system over t time slots are given by $\Gamma_t^m = \sum_{i=1}^{t}\gamma_i^m\tau$ and $\Gamma_t = \sum_{i=1}^{t}\sum_{m=1}^{M}\gamma_i^m\tau$, respectively. Additionally, the average throughput achieved by GU-m over previous t time slots is $\bar{\Gamma}_t^m = \Gamma_t^m/t$.

3.1.4 *Fairness-weighted throughput*

Blindly maximizing overall system throughput can achieve high spectral efficiency. However, it may lead to unfair communications among GUs, where only a small subset of GUs receives service in most time slots, leaving the rest constantly unserved. To address this issue, we introduce Jain's fairness index [47] to evaluate fairness among GUs. Specifically, we define a throughput fairness index based on each GU's average throughput over preceding time slots. The throughput fairness index at time slot t is calculated using the following equation:

$$\eta_t = \frac{(\sum_{m=1}^{M}\bar{\Gamma}_t^m)^2}{M(\sum_{m=1}^{M}(\bar{\Gamma}_t^m)^2)}. \tag{8}$$

According to Cauchy-Buniakowsky-Schwarz inequality, η_t always satisfies $0 \leqslant \eta_t \leqslant 1$. A higher fairness index indicates narrower variations in throughput among different GUs. The maximum value of η_t is reached when all GUs achieve the same overall throughput.

To strike a balance between maximizing the overall system throughput and ensuring a certain level of fairness, we construct a fairness-weighted throughput metric. This metric measures the cumulative throughput over the given period, using the fairness index as a weighting factor for each time slot. Mathematically, the formulation of fairness-weighted throughput is presented as follows:

$$\Gamma_f = \sum_{t=1}^{T}\eta_t\sum_{m=1}^{M}\gamma_t^m\tau. \tag{9}$$

3.2 Problem formulation

We aim to jointly optimize UAV trajectories, user association, and GUs' transmit power to maximize the fairness-weighted throughput. The corresponding optimization problem can be formulated as

$$\max_{\boldsymbol{q},\boldsymbol{\alpha},\boldsymbol{p}} \ \Gamma_f \quad \text{s.t. (1), (2), (4), (5),} \tag{10}$$

where $\boldsymbol{q} = \{\boldsymbol{q}_t^u | u\in\mathcal{U}, t\in\mathcal{T}\}$, $\boldsymbol{\alpha} = \{\alpha_t^{u,m} | u\in\mathcal{U}, m\in\mathcal{M}, t\in\mathcal{T}\}$, and $\boldsymbol{p} = \{p_t^m | m\in\mathcal{M}, t\in\mathcal{T}\}$ represent the variables for UAV trajectories, user association, and GUs' transmit power, respectively.

The optimization problem in (10) is an NP-hard mixed integer nonlinear programming problem, involving both continuous variables ($\boldsymbol{q}$ and $\boldsymbol{p}$) and discrete variables ($\boldsymbol{\alpha}$). The interdependence among these variables results in an extensive solution space. Moreover, the non-convex nature of the objective function and constraints adds the complexity. In addition to these challenges, several practical constraints must be considered.

(1) Centralized control is not feasible in a multi-UAV-assisted communication system due to high feedback costs and the absence of central controllers. Therefore, optimization should be carried out in a decentralized manner, with each UAV and GU making individual controlling decisions. Specifically, UAVs must design their own trajectories, while GUs have to determine their optimal user association and transmit power levels.

(2) Information exchange between UAVs and GUs, as well as among different UAVs, is permissible through control channels. However, we assume that different GUs are prohibited from exchanging control messages with each other due to privacy concerns.

(3) Limited sensing and communication abilities constrain each UAV and GU to partial information from neighbors. For clarity, we define two time-varying sets for each UAV and GU to represent its neighboring UAVs and GUs:

$$\begin{cases} \mathcal{U}_t^{\xi,i} = \{u | u \in \mathcal{U}, d_t^{\xi,i,u} \leqslant R_{\text{sense}}^{\xi}\}, \\ \mathcal{M}_t^{\xi,i} = \{m | m \in \mathcal{M}, d_t^{\xi,i,m} \leqslant R_{\text{sense}}^{\xi}\}, \end{cases} \quad \xi \in \{\text{uav}, \text{user}\}, \tag{11}$$

where i denotes the UAV or GU index, $d_t^{\xi,i,j}$ is the horizontal distance between UAV/GU i and UAV/GU j, and R_{sense}^{ξ} represents UAVs/GUs' sensing range.

(4) Real-time optimization is essential to accommodate the changing network topology and channel conditions caused by UAV movement.

To address these challenges, we recast this problem into a cooperative Dec-POMDP and employ MADRL to solve it.

4 Dec-POMDP formulation

A Dec-POMDP is an extension of the Markov decision process, designed to accommodate multiple agents [37]. At each time slot, the environment information is described by state $\boldsymbol{s}_t \in \boldsymbol{S}$. Each agent $i \in \mathcal{I}$ receives an individual observation $\boldsymbol{o}_t^i \in \mathcal{O}^i$, which provides partial information about the state $\boldsymbol{s}_t$. Subsequently, each agent selects an action $a^i \in \mathcal{A}^i$, guided by a policy $\pi^i(a^i|\boldsymbol{o}^i) : \mathcal{O}^i \times \mathcal{A}^i \mapsto [0,1]$. The joint action $\boldsymbol{a} \in \mathcal{A}^1 \times \cdots \times \mathcal{A}^I$ taken by all agents results in a transition to the next state $\boldsymbol{s}_{t+1}$. Meanwhile, all agents receive a shared reward r_t. The term "return" refers to the cumulative reward, i.e,

$$R_t = \sum_{j=0}^{\infty} \gamma^j r_{t+j},$$

where $\gamma \in [0,1)$ denotes the discount factor.

We consider UAVs and GUs as two types of heterogeneous agents, each possessing distinct observation and action spaces. All agents make decisions by the end of each time slot, ensuring that actions related to trajectory control or resource allocation can be executed in the subsequent time slot. Following the optimization problem described in Section 3, we define the essential components of our formulated Dec-POMDP as follows:

(1) Observation. Owing to partial observability, each agent is restricted to perceiving environmental information within its designated sensing range. The observation of each UAV agent $u \in \mathcal{U}$ includes its own current absolute coordinate, the relative coordinates of all neighboring UAVs and GUs, the average throughput over the last t slots, and the transmit power during the preceding time slot of all neighbor GUs, i.e,

$$\boldsymbol{o}_t^{\text{uav},u} = \left(\boldsymbol{q}_t^u, \left\{ \tilde{\boldsymbol{q}}_t^{u'} \right\}_{u' \in \mathcal{U}_t^u}, \{\tilde{\boldsymbol{w}}_t^m\}_{m \in \mathcal{M}_t^u}, \left\{ \bar{\Gamma}_t^m \right\}_{m \in \mathcal{M}_t^u}, \left\{ p_{t-1}^m \right\}_{m \in \mathcal{M}_t^u} \right). \tag{12}$$

For each GU agent $m \in \mathcal{M}$, its observation comprises its own current absolute coordinate, the ID of its last associated UAV, its transmit power at the last time slot, and the relative coordinates of all its neighbor UAVs, which is given by

$$\boldsymbol{o}_t^{\text{user},m} = \left(\boldsymbol{w}_t^m, u_{t-1}^m, p_{t-1}^m, \{\boldsymbol{q}_t^u\}_{u \in \mathcal{U}_t^m} \right), \tag{13}$$

where $u_t^m \in \mathcal{U}' \equiv \{0\} \bigcup \mathcal{U}$ denotes the ID of GU-m's associated UAV at time slot t and $u_t^m = 0$ represents the absence of an associated UAV.

(2) State. In [22], the concatenation of all observations is used as the environment state. Despite its generality, this method encounters scalability issues when dealing with a large number of agents. As an alternative, we form the state by constructing a vector containing information about all agents, i.e,

$$\boldsymbol{s}_t = \left(\left\{\boldsymbol{q}_t^u\right\}_{u\in\mathcal{U}}, \left\{\boldsymbol{w}_t^m\right\}_{m\in\mathcal{M}}, \left\{u_{t-1}^m\right\}_{m\in\mathcal{M}}, \left\{p_{t-1}^m\right\}_{m\in\mathcal{M}}, \left\{\bar{\Gamma}_t^m\right\}_{m\in\mathcal{M}}\right). \tag{14}$$

(3) Action. At each time slot, UAVs have to adjust their positions to provide fair service. Therefore, the action of UAV agent $u \in \mathcal{U}$ is defined as

$$a_t^{\mathrm{uav},u} = \varphi_t^u, \tag{15}$$

where φ_t^u denotes the UAV flying direction at time slot t. For GU agents, the task involves not only selecting associated UAVs but also determining the transmission power. Accordingly, the action of GU agent $m \in \mathcal{M}$ is defined as

$$a_t^{\mathrm{user},m} = \left(u_t^m, p_t^m\right). \tag{16}$$

The action space for UAV agents is continuous, while that for GU agents is hybrid. Since searching for optimal actions in continuous space remains challenging for MADRL configuration, we discretize all continuous actions. Specifically, the continuous flying direction is discretized into a finite set, namely, $\varphi_t^u \in \{-\pi/2, 0, \pi/2, \pi\}$. Similarly, the range of GUs' transmit power $[0, P_{\max}]$ is quantized into b power levels, denoted by $p_t^m \in \{0, P_{\max}/(b-1), 2P_{\max}/(b-1), \ldots, P_{\max}\}$. The discrete power levels and the set of associated UAVs' IDs $\mathcal{U}'$ constitute the joint action space for GU agents. As a result, the action spaces for both UAV and GU agents become discrete. Their dimensions are given by $|\mathcal{A}^{\mathrm{uav}}| = 4$ and $|\mathcal{A}^{\mathrm{user}}| = b(U+1)$, respectively.

(4) Reward. Our objective is to maximize total fairness-weighted throughput throughout the entire service duration. To achieve this, the reward is defined as the fairness-weighted throughput at each time slot. Additionally, we incorporate an extra penalty term to prevent collisions among UAVs. Therefore, the reward is expressed as

$$r_t = \omega^{\mathrm{r}} \eta_t \sum_{m=1}^{M} \gamma_t^m \tau + \omega^{\mathrm{c}} \delta_t^{\mathrm{collide}}, \tag{17}$$

where $\delta_t^{\mathrm{collide}}$ denotes the count of collisions brought about by all UAVs during time slot t. ω^{r} and ω^{c} represent scaling factors, serving two main functions. Firstly, they scale reward values to a range from -10 to 10, which is essential for effective neural network training. Secondly, they balance the relative importance of the optimization objective term and the collision penalty term within the reward function. The specific values of ω^{r} and ω^{c} are determined through empirical analysis.

Based on this formulation, the goal of MADRL is to find an optimal joint policy $\boldsymbol{\pi}^* = \{\pi^i\}_{i\in\mathcal{I}}$ such that the expected return $J = \mathbb{E}_{\boldsymbol{a}_{0:\infty}\sim\boldsymbol{\pi}, \boldsymbol{s}_{1:\infty}\sim P}[R_0]$ can be maximized. Typically, the optimal Q-function is used to describe the expected return when starting in state $\boldsymbol{s}_t$, taking an action $\boldsymbol{a}_t$, and thereafter following the optimal policy π^*:

$$Q^*(\boldsymbol{s}_t, \boldsymbol{a}_t) = \mathbb{E}_{\boldsymbol{a}_{n:\infty}\sim\boldsymbol{\pi}^*, \boldsymbol{s}_{n:\infty}\sim P}[R_t|\boldsymbol{s}_t, \boldsymbol{a}_t]. \tag{18}$$

Given $Q^*(\boldsymbol{s}_t, \boldsymbol{a}_t)$, the optimal policy π^* can be derived by selecting the greedy action:

$$\boldsymbol{a}_t^* = \arg\max_{\boldsymbol{a}_t} Q^*(\boldsymbol{s}_t, \boldsymbol{a}_t). \tag{19}$$

In single-agent DRL, the well-known DQN [23] algorithm approximates $Q^*(\boldsymbol{s}_t, \boldsymbol{a}_t)$ with a DNN function $Q_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t)$ parameterized by θ and trains it using samples stored in the experience replay buffer.

Learning optimal policies in multi-agent settings is considerably more complex than in single-agent cases. One approach trains a centralized Q-function that relies on the global state and joint action by extending DQN to multi-agent settings. However, the training of this function becomes challenging as the number of agents increases, and it lacks the capability to generate local policies for distributed execution. An alternative method trains each agent independently, treating other agents as part of the environment. This approach often fails to converge due to the non-stationarity problem, as the environment of each agent evolves alongside policy updates of others. Another critical challenge faced by both centralized and independent DQN is the multi-agent credit assignment issue: in fully cooperative settings with only

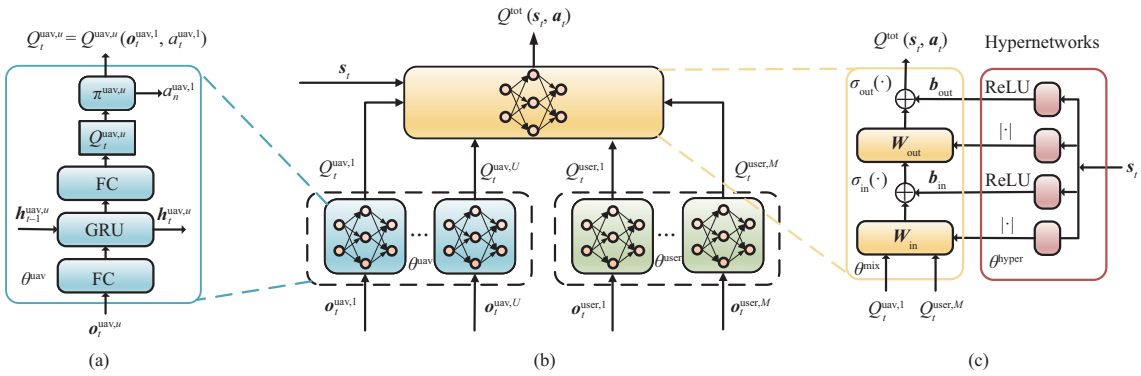

Figure 2 (Color online) Diagram of the proposed HC-QMIX architecture. (a) The local Q-networks take observations as input and generate individual Q-values; (b) the proposed HC-QMIX consists of multiple local Q-networks and a shared mixing network; (c) the shared mixing network combines individual Q-values into the joint Q-value. The weights and biases of the mixing network are produced by a series of hypernetworks.

team reward, it becomes difficult for agents to determine their individual contributions to the team's success [36]. Consequently, learning can be easily trapped in inefficient policies, where only a small portion of agents is active and the others are "lazy". This issue is particularly challenging in our scenario due to the varying capabilities, roles, and contributions of UAV and GU agents. Next, we will introduce a novel heterogeneous MADRL framework that can address the aforementioned challenges.

5 Proposed HC-QMIX algorithm

In this section, we present the proposed HC-QMIX algorithm, which can learn local policies for both UAV and GU agents in a coordinated manner. The fundamental concept of HC-QMIX involves estimating the joint Q-value as a complex non-linear combination of individual Q-values of heterogeneous agents that condition only on local observations. This process is referred to as heterogeneous value factorization. By ensuring the monotonicity of this combination function, the global optimum can be achieved by each agent selecting greedy actions based on local Q-values.

To realize this approach, HC-QMIX comprises a set of local Q-networks to learn individual Q-values for UAV and GU agents, and a shared mixing network combining these Q-values into the joint Q-value. The overall architecture of HC-QMIX is illustrated in Figure 2. In the following, we will provide a comprehensive description of both the local Q-networks and the mixing network.

5.1 Local Q-network

As depicted in Figure 2(a), a local Q-network represents the individual Q-function $Q^{\xi,i}(\boldsymbol{o}_t^{\xi,i}, a_t^{\xi,i})$ for each agent (ξ, i), where $\xi \in \{\text{uav}, \text{user}\}$ and i denotes the UAV/GU index. It takes the agent's observation as input and generates Q-values for all actions in the action space. Based on these Q-values, the optimal action and its corresponding Q-value are determined by using an arbitrary policy.

Owing to partial observability, estimating a Q-function from partial environmental information can be highly inaccurate, i.e, $Q^{\xi,i}(\boldsymbol{o}_t^{\xi,i}, a_t^{\xi,i}) \neq Q^{\xi,i}(\boldsymbol{s}_t, a_t^{\xi,i})$. To address this issue, we construct a three-layer network consisting of two fully-connected (FC) layers and a gate recurrent unit (GRU). Specifically, the input FC layer processes the observations $\boldsymbol{o}_t^{\xi,i}$ into an embedding feature $\boldsymbol{z}_t^{\xi,i}$. The GRU layer maintains a hidden state $\boldsymbol{h}_t^{\xi,i}$ determined by the current embedding feature $\boldsymbol{z}_t^{\xi,i}$ and the previous hidden state $\boldsymbol{h}_{t-1}^{\xi,i}$. This structure incorporates memory into local Q-networks, improving the estimation of environment dynamics and thereby reducing the discrepancy between $Q^{\xi,i}(\boldsymbol{o}_t^{\xi,i}, a_t^{\xi,i})$ and $Q^{\xi,i}(\boldsymbol{s}_t, a_t^{\xi,i})$. The output FC layer takes the current hidden state $\boldsymbol{h}_t^{\xi,i}$ as input and generates Q-values for all actions, i.e, $\boldsymbol{Q}_t^{\xi,i} = \{Q^{\xi,i}(\boldsymbol{o}_t^{\xi,i}, a^{\xi,i}) | a^{\xi,i} \in \mathcal{A}^{\xi}\}$. Finally, we utilize these Q-values to determine the optimal action $a_t^{\xi,i}$ and its Q-value $Q_t^{\xi,i} = Q^{\xi,i}(\boldsymbol{o}_t^{\xi,i}, a_t^{\xi,i})$. For example, the ϵ-greedy polices can be expressed as

$$\pi^{\xi,i}(a_t^{\xi,i}|\boldsymbol{o}_t^{\xi,i}) = \begin{cases} 1-\epsilon+\frac{\epsilon}{|\mathcal{A}^{\xi}|}, & \text{if } a_t^{\xi,i} = \arg\max_{a^{\xi,i}} Q^{\xi,i}(\boldsymbol{o}_t^{\xi,i}, a^{\xi,i}), \\ \frac{\epsilon}{|\mathcal{A}^{\xi}|}, & \text{otherwise.} \end{cases} \tag{20}$$

Unlike QMIX that optimizes a shared local Q-network for all agents, HC-QMIX learns a separate local Q-network for each agent type. This means that each type of agents possesses its own dedicated Q-network to capture their unique dynamics, facilitating better decision-making. For homogeneous agents, since they have identical action and observation space as well as reward function, a single Q-network is shared. This avoids an exponential increase in learnable parameters as the number of agents grows. Let θ^{uav} and θ^{user} denote the parameters of Q-networks for UAV agents and GU agents, respectively. Consequently, their local Q-networks can be represented by $Q_{\theta^{\text{uav}}}^{\text{uav},u}\left(\boldsymbol{o}_t^{\text{uav},u}, a_t^{\text{uav},u}\right)$ and $Q_{\theta^{\text{user}}}^{\text{user},m}\left(\boldsymbol{o}_t^{\text{user},m}, a_t^{\text{user},m}\right)$ for each $u \in \mathcal{U}$ and $m \in \mathcal{M}$. Despite sharing the same network, agents of the same type can still exhibit diverse behaviors with distinct local observations as input.

5.2 Shared mixing network

As shown in Figure 2(c), the shared mixing network represents the combination function f, which captures the relationship between the joint Q-value $Q^{\text{tot}}(\boldsymbol{s}_t, \boldsymbol{a}_t)$ and individual Q-values $Q^{\xi,i}(\boldsymbol{o}_t^{\xi,i}, a_t^{\xi,i})$. It takes the outputs of all local Q-networks as input and generates the joint Q-value. The structure of the mixing network comprises an FC input layer and an FC output layer. The input layer takes the individual Q-values as input and produces an embedding feature:

$$\boldsymbol{z}_t^{\text{mix}} = \sigma_{\text{in}}\left(\boldsymbol{W}_{\text{in}}\boldsymbol{Q}_t + \boldsymbol{b}_{\text{in}}\right), \tag{21}$$

where $\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{b}_{\text{in}}$ are the weight matrix and bias vector of the input layer, respectively. The activation function $\sigma_{\text{in}}(\cdot)$ is applied element-wise, and $\boldsymbol{Q}_t = [Q_t^{\text{uav},1}, \ldots, Q_t^{\text{uav},U}, Q_t^{\text{user},1}, \ldots, Q_t^{\text{user},M}]$ forms a vector of all individual Q-values. The output layer generates the joint Q-value by utilizing the embedding feature:

$$Q^{\text{tot}}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \sigma_{\text{out}}\left(\boldsymbol{W}_{\text{out}}\boldsymbol{z}_t^{\text{mix}} + \boldsymbol{b}_{\text{out}}\right), \tag{22}$$

where $\boldsymbol{W}_{\text{out}}$ and $\boldsymbol{b}_{\text{out}}$ are the weight matrix and bias vector of the output layer, and $\sigma_{\text{out}}(\cdot)$ represents the output activation function.

Ensuring consistency between the centralized policy and individual policies requires that the global arg max operation on the joint Q-value yields the same result as a sequence of arg max operations on each local Q-value for both UAV and GU agents. This relationship can be concisely expressed as follows:

$$\arg\max_{\boldsymbol{a}} Q^{\text{tot}}\left(\boldsymbol{s}_t, \boldsymbol{a}\right) = \begin{pmatrix} \arg\max_{a^{\text{uav},1}} Q^{\text{uav},1}\left(\boldsymbol{o}_t^{\text{uav},1}, a^{\text{uav},1}\right) \\ \vdots \\ \arg\max_{a^{\text{uav},U}} Q^{\text{uav},U}\left(\boldsymbol{o}_t^{\text{uav},U}, a^{\text{uav},U}\right) \\ \arg\max_{a^{\text{user},1}} Q^{\text{user},1}\left(\boldsymbol{o}_t^{\text{user},1}, a^{\text{user},1}\right) \\ \vdots \\ \arg\max_{a^{\text{user},M}} Q^{\text{user},M}\left(\boldsymbol{o}_t^{\text{user},M}, a^{\text{user},M}\right) \end{pmatrix}. \tag{23}$$

When Eq. (23) holds, each agent can attain the global optimum by selecting the action that maximizes their individual Q-value. This constraint is met by imposing the condition of monotonicity on the function f:

$$\frac{\partial Q^{\text{tot}}(\boldsymbol{s}_t, \boldsymbol{a}_t)}{\partial Q^{\xi,i}(\boldsymbol{o}_t^{\xi,i}, a_t^{\xi,i})} \geqslant 0, \ \forall i. \tag{24}$$

To enforce this monotonicity, the weights ($\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{W}_{\text{out}}$) of the shared mixing network must be non-negative. This objective is achieved through a set of hypernetworks responsible for weight generation [48]. As shown in the right part of Figure 2(c), each hypernetwork, composed of a linear layer and an absolute activation function, outputs the weight matrix for one layer in the mixing network, using the state $\boldsymbol{s}_t$ as input. The bias vectors are generated similarly, but replace the absolute activation function with the ReLU activation. The environment state $\boldsymbol{s}_t$ serves as additional information to facilitate learning. Notably, $\boldsymbol{s}_t$ is fed into hypernetworks, not the mixing network, since the latter can solely represent monotonic functions. This approach permits non-monotonic conditioning of $Q^{\text{tot}}(\boldsymbol{s}_t, \boldsymbol{a}_t)$ on $\boldsymbol{s}_t$.

Let $g_{\theta^{\text{hyper}}}(\cdot)$ denote the function represented by hypernetworks, with θ^{hyper} representing the network parameters. Then, the parameters of the mixing network, comprising weight matrices ($\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{W}_{\text{out}}$)

Algorithm 1 Learning process of HC-QMIX

Input: Initial UAV Q-network parameters θ^{uav}, GU Q-network parameters θ^{user}, hypernetwork parameters θ^{hyper}, and empty replay buffer $\mathcal{D}$;
Set target parameters equal to main parameters $\theta^{\text{uav}-} \leftarrow \theta^{\text{uav}}$, $\theta^{\text{user}-} \leftarrow \theta^{\text{user}}$, $\theta^{\text{hyper}-} \leftarrow \theta^{\text{hyper}}$;
repeat
  Reset environment state;
  for $i \in \mathcal{I}$ **do**
    Initialize the hidden state $\boldsymbol{h}_t^i$ as all-zero vector;
  end for
  for $t = 1, \ldots, T$ **do**
    Obtain the environment state $\boldsymbol{s}_t$;
    for $i \in \mathcal{I}$ **do**
      Observe $\boldsymbol{o}_t^i$ and select action a_t^i following (20);
      Execute a_t^i in the environment;
      Update the hidden state $\boldsymbol{h}_t^i$;
    end for
    for $i \in \mathcal{I}$ **do**
      Obtain the next observation $\boldsymbol{o}_{t+1}^i$;
    end for
    Obtain the next state $\boldsymbol{s}_{t+1}$ and team reward r_t;
  end for
  Store $\{(\boldsymbol{s}_t, \{\boldsymbol{o}_t^i\}_{i\in\mathcal{I}}, \{a_t^i\}_{i\in\mathcal{I}}, \boldsymbol{s}_{t+1}, \{\boldsymbol{o}_{t+1}^i\}_{i\in\mathcal{I}}, r_t, \{\boldsymbol{h}_t^i\}_{i\in\mathcal{I}})\}_{t=1:T}$ in replay buffer $\mathcal{D}$;
  if It is time to update **then**
    Randomly sample a batch of episodes $\mathcal{B} = \{\{(\boldsymbol{s}_t, \{\boldsymbol{o}_t^i\}_{i\in\mathcal{I}}, \{a_t^i\}_{i\in\mathcal{I}}, \boldsymbol{s}_{t+1}, \{\boldsymbol{o}_{t+1}^i\}_{i\in\mathcal{I}}, r_t, \{\boldsymbol{h}_t^i\}_{i\in\mathcal{I}})\}_{t=1:T}\}$ from $\mathcal{D}$;
    Update $\boldsymbol{\theta} = \{\theta^{\text{uav}}, \theta^{\text{user}}, \theta^{\text{hyper}}\}$ according to (26);
    if It is time to update target parameters **then**
      Update $\boldsymbol{\theta}^-$ with $\theta^{\text{uav}-} \leftarrow \theta^{\text{uav}}$, $\theta^{\text{user}-} \leftarrow \theta^{\text{user}}$, $\theta^{\text{hyper}-} \leftarrow \theta^{\text{hyper}}$;
    end if
  end if
until Convergence;
Output: Well-trained UAV Q-network parameters $\theta^{\text{uav}*}$ and GU Q-networks parameters $\theta^{\text{user}*}$.

and bias vectors ($\boldsymbol{b}_{\text{in}}$ and $\boldsymbol{b}_{\text{out}}$), can be calculated as $\theta^{\text{mix}} = g_{\theta^{\text{hyper}}}(\boldsymbol{s}_t)$. Based on this, the global Q-value Q^{tot} can be expressed as

$$Q_{\boldsymbol{\theta}}^{\text{tot}}(\boldsymbol{s}_t, \boldsymbol{a}_t) = f_{\theta^{\text{mix}}}\left(Q_{\theta^{\text{uav}}}^{\text{uav},1}, \ldots, Q_{\theta^{\text{user}}}^{\text{user},M}\right), \tag{25}$$

where $\boldsymbol{\theta} = \left\{\theta^{\text{uav}}, \theta^{\text{user}}, \theta^{\text{hyper}}\right\}$ denotes the collection of all network parameters that concurrently determine the value of Q^{tot}.

5.3 Learning process

We adopt the framework of centralized training and distributed execution in our approach.

5.3.1 *Centralized training*

During the training process, agents gather experience by interacting with the environment using ϵ-greedy policies derived from local Q-networks. The interactions between agents and the environment are organized into episodes of uniform length T. At the beginning of each episode, the environment state is reset, and the hidden state $\boldsymbol{h}_t^i$ of each agent-i's local Q-network is initialized as an all-zero vector. The set of transition tuples $\{(\boldsymbol{s}_t, \{\boldsymbol{o}_t^i\}_{i\in\mathcal{I}}, \{a_t^i\}_{i\in\mathcal{I}}, \boldsymbol{s}_{t+1}, \{\boldsymbol{o}_{t+1}^i\}_{i\in\mathcal{I}}, r_t, \{\boldsymbol{h}_t^i\}_{i\in\mathcal{I}})\}_{t=1:T}$ collected from each episode is stored as experience in the replay buffer $\mathcal{D}$. Periodically, we sample a batch of episodes $\mathcal{B}$ from $\mathcal{D}$ and train network parameters using the stochastic gradient ascent method [49]. Specifically, the parameters $\boldsymbol{\theta} = \left\{\theta^{\text{uav}}, \theta^{\text{user}}, \theta^{\text{hyper}}\right\}$ are updated end-to-end by minimizing the following loss:

$$\mathcal{L}_{Q^{\text{tot}}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathcal{B}}\left[y_t^{\text{tot}} - Q_{\boldsymbol{\theta}}^{\text{tot}}(\boldsymbol{s}_t, \boldsymbol{a}_t)\right], \tag{26}$$

where $y_t^{\text{tot}} = r_t + \gamma \max_{\boldsymbol{a}_{t+1}} Q_{\boldsymbol{\theta}^-}^{\text{tot}}(\boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1})$ represents the bootstrapping target of Q^{tot}, and $\boldsymbol{\theta}^-$ is periodically copied from $\boldsymbol{\theta}$. The complete training procedure is outlined in Algorithm 1.

5.3.2 *Distributed execution*

Once convergence is achieved, the well-trained local Q-networks can be directly deployed on each UAV/GU. The optimal policies are derived by selecting greedy actions as follows:

$$a_t^{\xi,i} = \arg\max_{a^{\xi,i}} Q^{\xi,i}\left(\boldsymbol{o}_t^{\xi,i}, a^{\xi,i}\right). \tag{27}$$

Table 2 Main environment parameters

Parameter	Value	Unit
Duration of each time slot (τ)	10	s
Flying altitude of UAVs (H)	100	m
Flying speed of UAVs (V)	10	m/s
Minimum distance between UAVs	5	m
Bandwidth of the spectrum (B)	10	MHz
Channel gain at the reference distance of 1 m (ρ_0)	-50	dB
PSD of AWGN (N_0)	-169	dBm/Hz
Sensing range for UAVs/GUs ($R_{\text{sense}}^{\text{uav}}/R_{\text{sense}}^{\text{user}}$)	500	m

This enables distributed and real-time implementation.

In summary, the proposed HC-QMIX algorithm offers several benefits. On the one hand, training of the centralized Q-function mitigates the non-stationarity problem by utilizing global environmental information. On the other hand, the shared mixing network decomposes the joint Q-value, representing the expected return for all agents, into individual Q-values that correspond to the contributions of individual UAVs and GUs. This decomposition enables HC-QMIX to allocate credit more accurately to each agent's actions, addressing the multi-agent credit assignment issue. Moreover, the coordination between UAV and GU agents is substantially promoted via the shared mixing network. After sufficient training, the factored local Q-networks allow each agent to make decisions independently, facilitating distributed online implementation.

5.4 Complexity analysis

The training can be carried out offline, while execution occurs online at each agent. Therefore, we focus on evaluating the implementation complexity for individual agents, which depends on the forward computation of well-trained local Q-networks. The architecture of a local Q-network comprises three components: an FC input layer, a GRU middle layer, and an FC output layer. The main operation in an FC layer involves multiplying the input vector by the weight matrix and adding a bias term. Its computational complexity is $\mathcal{O}(d_{\text{in}} \cdot d_{\text{out}})$ [50], with d_{in} and d_{out} representing the dimensions of input and output vector, respectively. The key computations within a GRU layer entail two matrix multiplications and a series of element-wise operations, including reset gate, update gate, and hidden state gate. Its computational complexity is $\mathcal{O}((d_{\text{in}} + d_{\text{hidden}}) \cdot d_{\text{hidden}})$ [51], where d_{hidden} represents the dimension of hidden states. To sum up, the overall computational complexity of a local Q-network is $\mathcal{O}(d_{\text{obs}} \times d_{\text{emb}} + (d_{\text{emb}} + d_{\text{hidd}}) \cdot d_{\text{hidd}} + d_{\text{hidd}} \times |\mathcal{A}|)$, where d_{obs}, d_{emb}, and d_{hidd} denote the dimensions of the observation vector, embedding feature, and hidden state, respectively, and $|\mathcal{A}|$ represents the total number of actions in the action space.

6 Simulation results

In this section, we first compare HC-QMIX with state-of-the-art benchmark algorithms. Next, we conduct a comparative analysis between the joint UAV trajectory design, user association, and power control (TAPC) optimization scheme and three benchmark communication schemes. Finally, we evaluate the performance of the proposed algorithm across various experiment setups.

6.1 Simulation settings

The service region of UAVs is restricted to a square area of 2000 m $\times$ 2000 m. At the beginning of each episode, UAV and GU locations are uniformly generated within this area. We set the number of UAVs and GUs as $U = 4$ and $M = 5$, and the maximum transmit power of GUs as $P_{\max} = 0.2$ W. The quantization precision of transmit power is set to $b = 5$, meaning that p_t^m is limited to the set $\{0, 0.05 \text{ W}, 0.1 \text{ W}, 0.15 \text{ W}, 0.2 \text{ W}\}$. In Subsection 6.4, we will adjust the settings of U, M, and $P_{\max}$ to validate the robustness of the proposed algorithm. Other main environment parameters are summarized in Table 2.

Regarding the algorithm, we set the dimensions of both embedding features and hidden states in the local Q-network as 128. During training, the value of ϵ is linearly annealed from 1 to 0.05 over 200000

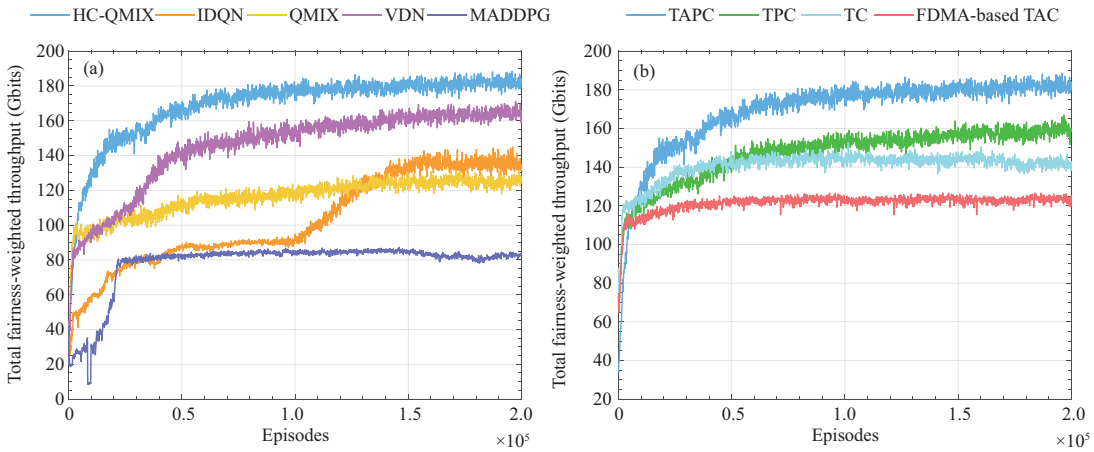

Figure 3 (Color online) Convergence comparison in terms of total fairness-weighted throughput per episode. (a) Comparison among different MADRL algorithms; (b) comparison among different communication schemes.

steps and then remains fixed until training ends. In all experiments, γ is set to 0.99. We adopt Adam optimizer to update network parameters with adaptive learning rates. The initial learning rate is set as 0.0005. The replay buffer stores the most recent 2000 episodes. Each update uses a batch of 32 episodes from the replay buffer. The agents are trained over 200000 episodes, with each episode including 150 time steps within the environment.

6.2 Performance comparison among MADRL algorithms

To demonstrate the superiority of HC-QMIX, we adopt four MADRL algorithms as benchmarks:

Independent DQN (IDQN). Similar to [29], the policy of each agent is trained independently based on the DQN algorithm [23].

MADDPG. Following [44], agents are trained utilizing the MADDPG algorithm [22], where a centralized critic and a local actor are trained for each agent within the framework of centralized training with distributed execution.

Value-decomposition network (VDN). Similar to [52], agents are trained using VDN algorithm [36]. This algorithm represents the joint Q-value as a sum of individual Q-values, which are estimated by a set of local Q-networks.

QMIX. UAV and GU agents are trained simultaneously but separately based on vanilla QMIX method [37], where an individual mixing network is trained for each group of agents.

Figure 3(a) presents the convergence curve for total fairness-weighted throughput. During the initial stage of training, IDQN is trapped in a local optimum, ceasing progress after reaching a mere 50% of the performance achieved by HC-QMIX. This is attributed to the inherent non-stationarity in independent learning. As training progresses, agents progressively reduce exploration and their policies tend to be stable, alleviating non-stationarity in the environment. Consequently, the performance of IDQN steadily improves and eventually approaches 75% of that achieved by HC-QMIX. Surprisingly, MADDPG exhibits the poorest performance, approximately 50% lower than HC-QMIX. In MADDPG, a centralized critic is trained for each agent, which receives global state-action pairs as input and outputs the global Q-value. Although this approach helps mitigate the non-stationarity issue through the use of global environmental information, it fails to address the multi-agent credit assignment problem. This is due to the lack of a mechanism to derive local Q-values for evaluating individual agents' contributions rather than the whole team. This explains why MADDPG performs the worst in our cooperative particularly observable TAPC problem. Value factorization tackles this issue by decomposing the global Q-value into agent-wise Q-values [37], which encourages coordination between agents. QMIX achieves a notable improvement over MADDPG by separately adopting value factorization for each group of agents. However, its performance is still lower than that of HC-QMIX. This is because QMIX only enables coordination within the groups of UAVs and GUs. In fact, the coordination between UAVs and GUs is also essential, as the UAV trajectory and GUs' transmit power, as well as user association, are deeply coupled with each other. To address this, HC-QMIX combines the Q-values of all agents into a shared mixing network, effectively approximating the coupling relationship between UAVs and GUs and promoting air-ground cooperation.

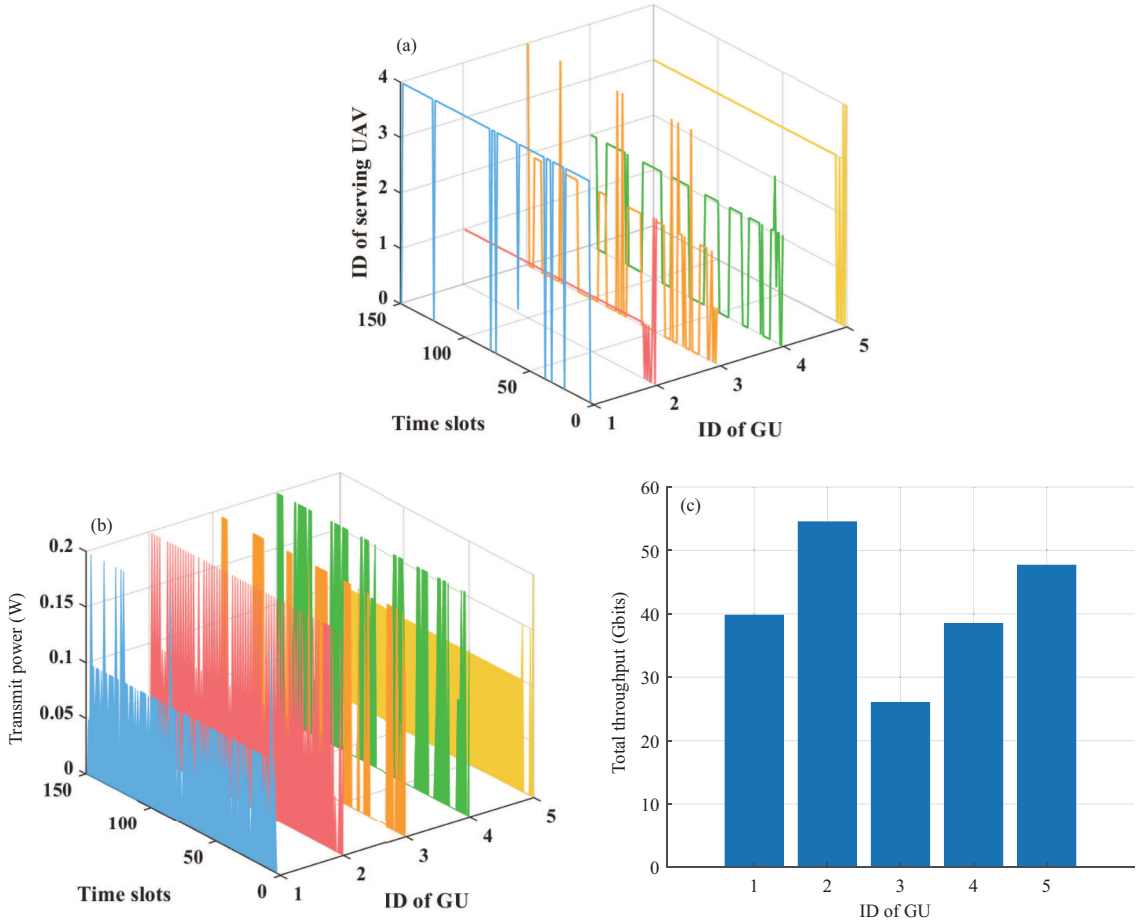

Figure 4 (Color online) Results obtained by TAPC scheme. (a) User association; (b) power control; (c) total throughput obtained by each GU over a single episode.

Different from HC-QMIX, VDN employs linear value factorization by representing the joint Q-value as a sum of individual Q-values. While this approach is adequate, it is not necessary for satisfying the monotonicity constraint of value factorization. In contrast, HC-QMIX utilizes a shared mixing network to estimate the combination relationship between the joint Q-value and individual Q-values. Consequently, it possesses a higher representational capacity to accurately represent the joint Q-value whereas VDN cannot. This sets HC-QMIX apart from VDN, accounting for the performance gap between the two. In conclusion, the superior performance of HC-QMIX over benchmarks shows the benefits of heterogeneous value factorization in dealing with non-stationarity issues, multi-agent credit assignment, and air-ground coordination.

6.3 Performance comparison among communication schemes

In this subsection, the proposed TAPC scheme is compared with three benchmark communication schemes:

• **Trajectory control (TC).** Only UAVs are considered agents to optimize their trajectories, while GUs always associate with UAVs offering the best channel and transmit at maximum power.

• **Trajectory and power control (TPC).** UAV trajectory and GUs' transmit power are jointly optimized, employing the same user association strategy as the TC scheme.

• **Frequency division multiple access-based trajectory and user association control (FDMA-based TAC).** The available spectrum is divided into U equal bands, each occupied by a UAV. In this case, GUs transmit at maximum power without causing interference to others. Therefore, only UAV trajectory and user association are optimized.

The optimization problems of these schemes are constructed differently. To ensure a fair comparison, they are all solved based on the proposed HC-QMIX or the QMIX algorithm (for the TC scheme only) with

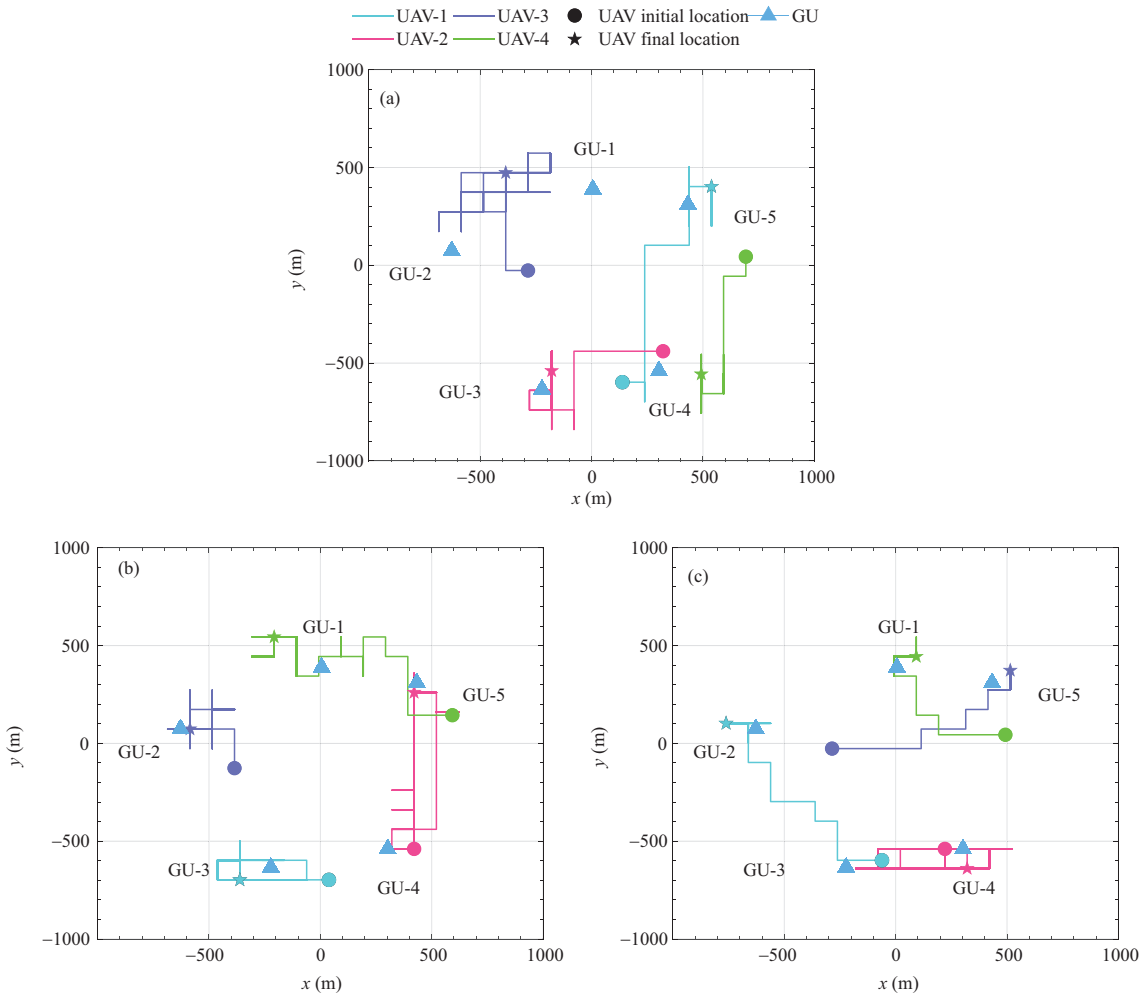

Figure 5 (Color online) Trajectories obtained by TC, TPC, and TAPC schemes. (a) Trajectories optimized by TC; (b) trajectories optimized by TPC; (c) trajectories optimized by TAPC.

identical hyperparameters, as detailed in Subsection 6.1. Therefore, at the agent level, the complexity of the three benchmark schemes is nearly equivalent to that of the proposed scheme. Their learning curves are shown in Figure 3(b).

As illustrated in Figure 3(b), the total fairness-weighted throughput achieved by the proposed TAPC scheme is approximately 182 Gbits after 200000 episodes, which is 26 Gbits, 38 Gbits, and 60 Gbits higher than that of TPC, TC, and FDMA-based TAC schemes, respectively. These results suggest that spectrum sharing with power control and user association yields a substantial performance gain. To elucidate the reasons, example results of user association and power control, as well as each GU's total throughput of a single episode are plotted in Figure 4. Given the same initial environment state, trajectories optimized by the TC, TPC, and TAPC schemes are presented in Figure 5. Several observations can be made from these results. Firstly, UAVs have learned to cooperatively provide fair service via trajectory design. For example, in Figure 5(c), UAV-1, UAV-3, and UAV-4 fly close to and hover around different GUs, to deliver high-quality communication services. Conversely, UAV-2 moves back and forth between GU-3 and GU-4, attempting to serve them alternatively. In this manner, both the overall throughput and fairness among GUs can be ensured. Secondly, power control offers greater flexibility for UAV trajectory design. It can be observed from Figure 5(a) that UAVs tend to maintain distance from one another when all GUs transmit at maximum power. This approach not only alleviates cross-link interference, but also sacrifices some direct communication links. On the contrary, in Figures 5(b) and (c), GUs' transmit power is jointly optimized with UAV trajectory. Even though some UAVs (e.g., UAV-3 and UAV-4 in Figure 5(c)) are close to each other, cross-link interference can still be mitigated by GUs adjusting their transmit power levels. As such, the potential benefits of UAV trajectory can be further exploited. Thirdly, power

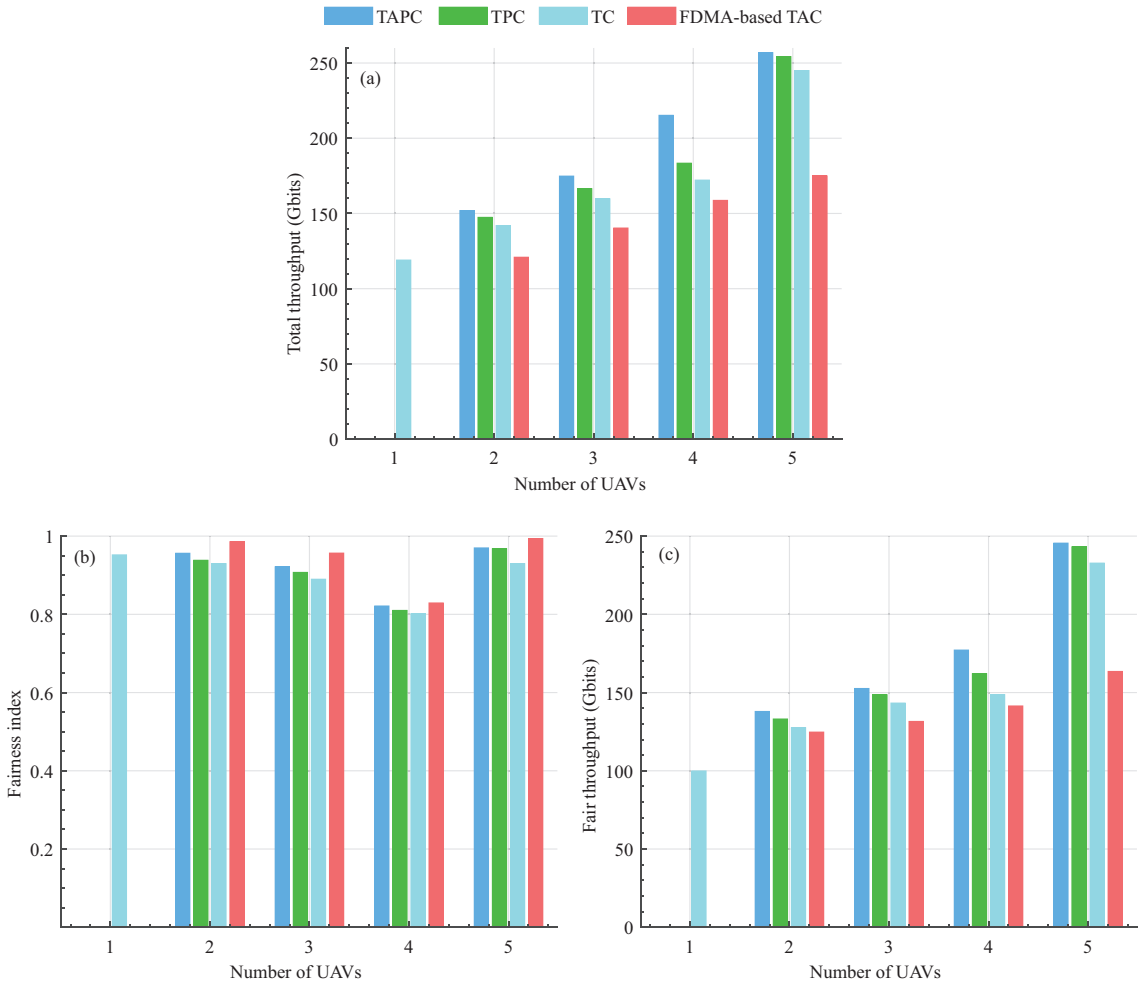

Figure 6 (Color online) Comparison of different metrics with respective to the number of UAVs. The number of GUs is set as 5. (a) Total throughput; (b) throughput fairness; (c) total fairness-weighted throughput.

control enhances the throughput fairness. As depicted in Figures 4(a) and (b), GU-1 and GU-5, which are continuously served, are more likely to communicate at low power levels (0.1 W). By contrast, GU-3 and GU-4, being disconnected for nearly half the period, almost always select maximum transmit power when being served. GU-2, constantly served but relatively distant from other GUs, switches its transmit power level between 0.05 and 0.2 W. With these flexible power adjustments, the throughput discrepancy among different GUs is reduced to a certain extent. Lastly, user association can further narrow the throughput gap. In Figure 4(a), since GU-3 has to compete with GU-4 for the service of UAV-2, it occasionally opts to associate with UAV-4 to increase the probability of being served. Simultaneously, GU-1, which is nearest to UAV-4, chooses not to associate with any UAV during these slots, ensuring that GU-3 can be scheduled by UAV-4. As a result, the overall throughput of GU-3 is enhanced, at the cost of GU-1's performance. Figure 4(c) displays the total throughput achieved by each GU. These results demonstrate that the throughput-fairness tradeoff is realized through spectrum sharing, power control, and user association.

6.4 Performance comparison across different experiment setups

In this subsection, the robustness of the proposed method is validated by comparing the aforementioned schemes across various experimental setups. The evaluation metrics employed include total throughput, fairness index, and total fairness-weighted throughput.

Figure 6 presents the performance comparison across varying numbers of UAVs, while keeping the number of GUs fixed at 5. In cases where there is only one UAV and cross-link interference is absent, power control and user association are unnecessary, and only the results of the TC scheme are provided.

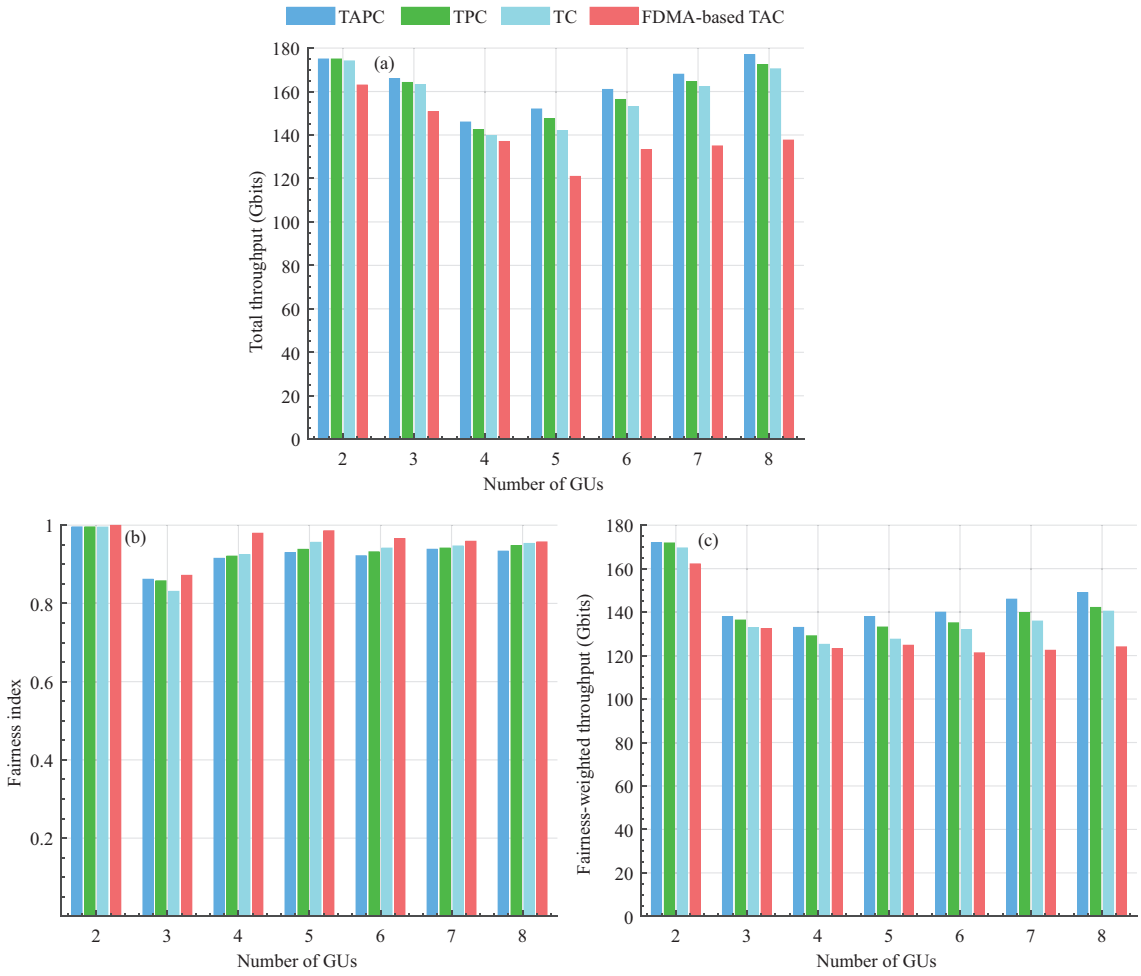

Figure 7 (Color online) Comparison of different metrics with respective to the number of GUs. The number of UAVs is set as 2. (a) Total throughput; (b) throughput fairness; (c) total fairness-weighted throughput.

As shown in Figure 6(a), the TC scheme achieves a total throughput of 120 Gbits in the single-UAV case. As the number of UAVs increases, this metric rises, reaching 249 Gbits when the number of UAVs is 5. Two primary factors contribute to the throughput improvement: (1) concurrent transmissions of multiple UAVs within the same spectrum budget significantly enhance spectrum efficiency, and (2) the average channel gain between UAVs and GUs improves as more UAVs cover the area. Concerning the various schemes, the TAPC scheme consistently outperforms others, highlighting the benefit of spectrum sharing and interference mitigation brought by power control and user association. Figure 6(b) reveals that the fairness index initially declines and subsequently increases for different schemes. In the beginning, scarce UAVs serve all GUs alternately. As the number of UAVs increases, some GUs receive continuous service while others are served intermittently, causing the drop of the fairness index. When the number of UAVs becomes adequate to serve all GUs, the fairness index rises again. In addition, the FDMA-based TAC scheme achieves the highest fairness index for all cases, followed by TAPC, TPC, and TC schemes. Nevertheless, in terms of total fairness-weighted throughput, TAPC, TPC, and TC schemes maintain a distinct advantage over the FDMA-based TAC scheme, with this advantage growing as the number of UAVs increases. In conclusion, employing multiple UAVs enhances the trade-off between total throughput and fairness index, particularly with effective interference management.

Figure 7 illustrates the results for varying numbers of GUs, while maintaining the number of UAVs at 2. With 2 GUs, each GU is assigned to a specific UAV, receiving consistent high-quality service. As the number of GUs increases, UAVs must traverse the entire area to serve GUs alternately, resulting in a decreased average channel gain between UAVs and GUs, as well as unbalanced services among GUs. This explains the initial decline in total throughput and fairness index for all schemes. When the number of GUs reaches a certain threshold, GUs become almost uniformly distributed across

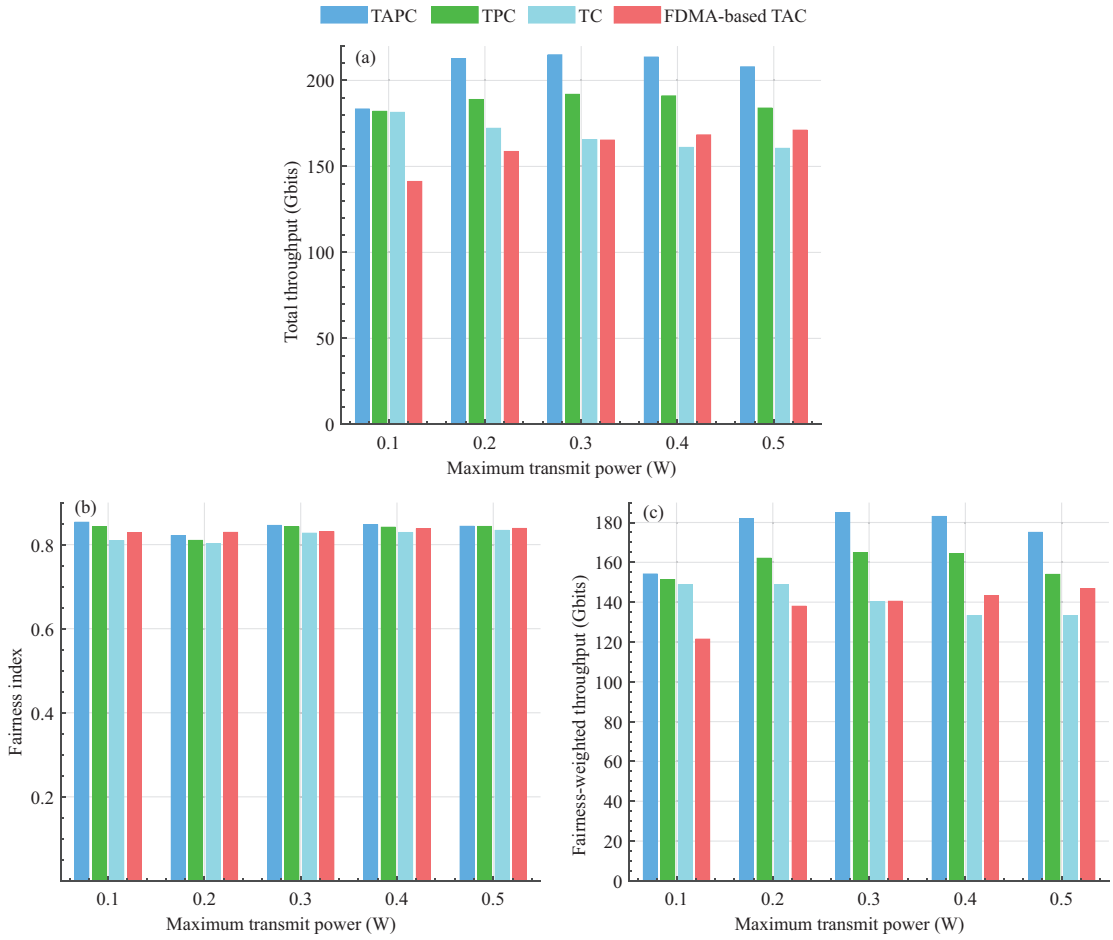

Figure 8 (Color online) Comparison of different metrics with respective to the maximum transmit power. The numbers of UAVs and GUs are set as 4 and 5, respectively. (a) Total throughput; (b) throughput fairness; (c) total fairness-weighted throughput.

the area, and the average channel gain rises once more. Consequently, total throughput increases steadily, and the fairness index stabilizes at approximately 0.95. While TAPC and TPC exhibit only a slight advantage over TC for the case with 2 GUs, their performance enhancement becomes more evident as the number of GUs increases. This occurs because communication resources are limited with a fixed number of UAVs, and as the number of GUs grows, power control and user association become increasingly important. Furthermore, TAPC, TPC, and TC schemes consistently outperform the FDMA-based TAC scheme, emphasizing the benefits of spectrum sharing coupled with efficient interference management.

Lastly, the impact of maximum transmit power on performance is examined. The number of UAVs and GUs are set as 4 and 5, respectively. As depicted in Figure 8, the total throughput achieved by the FDMA-based TAC scheme increases with maximum transmit power, as higher transmit power improves the data rate in interference-free settings. Conversely, the TC scheme experiences a noticeable decline in total throughput as maximum transmit power increases. This occurs because both direct-link transmission and cross-link interference intensify with rising transmit power. When the value of maximum transmit power increases from 0.1 to 0.5 W, the growth of cross-link interference surpasses that of direct-link transmission, resulting in decreased total throughput. In TAPC and TPC schemes, power control and user association mitigate cross-link interference to a certain extent, leading to a throughput advantage over the TC scheme. Furthermore, the larger the maximum transmit power, the greater the interference mitigation effectiveness. However, after an initial increase, the total throughput achieved by TAPC and TC stabilizes at a certain level rather than continuing to grow. This suggests that total throughput in interference channels reaches its maximum, and further increasing transmit power does not yield performance gains. Consequently, the advantages of TAPC, TPC, and TC schemes over the FDMA-based scheme diminish as the maximum transmit power increases. The maximum transmit power has a

minimal impact on the fairness index which remains around 0.84 for all schemes. Therefore, the results for total fairness-weighted throughput display similar trends to those for total throughput.

7 Conclusion

In this paper, we propose a novel heterogeneous MADRL approach to jointly optimize UAV trajectories, user association, and GUs' transmit power in multi-UAV-assisted communication systems. The proposed HC-QMIX algorithm tackles the multi-agent credit assignment issue through heterogeneous value factorization, and promotes coordination between UAVs and GUs via a shared mixing network. Particularly, our method enables distributed and real-time implementation, allowing each agent to make online decisions based on local observations. The proposed algorithm achieves significant performance improvement over benchmarks, and the effectiveness of the interference management scheme is further confirmed through extensive simulation results. Like most existing MADRL algorithms, HC-QMIX faces certain scalability issues and encounters challenges in scenarios with dynamic population sizes. Our future work may include the following aspects: (1) improving the learning efficiency of HC-QMIX by incorporating more efficient networks, such as Attention [53], DeepSets [54], and GNN [55]; (2) designing more scalable structures for settings with a dynamic number of agents; (3) integrating advanced adaptive learning rate mechanisms to further enhance convergence efficiency.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 62371462, 61931020, 62101569, U19B2024), Natural Science Foundation of Hunan Province (Grant No. 2022JJ10068), and Science and Technology Innovation Program of Hunan Province (Grant No. 2022RC1093).

References

1 Zeng Y, Zhang R, Lim T J. Wireless communications with unmanned aerial vehicles: opportunities and challenges. IEEE Commun Mag, 2016, 54: 36–42
2 Mozaffari M, Saad W, Bennis M, et al. A tutorial on UAVs for wireless networks: applications, challenges, and open problems. IEEE Commun Surv Tutorials, 2019, 21: 2334–2360
3 Zeng Y, Wu Q Q, Zhang R. Accessing from the sky: a tutorial on UAV communications for 5G and beyond. Proc IEEE, 2019, 107: 2327–2375
4 Wang H J, Zhao H T, Zhang J, et al. Survey on unmanned aerial vehicle networks: a cyber physical system perspective. IEEE Commun Surv Tutorials, 2020, 22: 1027–1070
5 Wang H J, Zhao H T, Ren B Q, et al. Cyber-physical framework for UAV intelligent communications (in Chinese). Sci Sin Inf, 2022, 52: 2041–2154
6 Zhao H T, Wang H J, Wu W Y, et al. Deployment algorithms for UAV airborne networks toward on-demand coverage. IEEE J Sel Areas Commun, 2018, 36: 2015–2031
7 Hentati A I, Fourati L C. Comprehensive survey of UAVs communication networks. Comput Standards Interfaces, 2020, 72: 103451
8 Wang H J, Jiang B, Zhao H T, et al. Joint resource allocation on slot, space and power towards concurrent transmissions in UAV ad hoc networks. IEEE Trans Wireless Commun, 2022, 21: 8698–8712
9 Zeng Y, Zhang R, Lim T J. Throughput maximization for UAV-enabled mobile relaying systems. IEEE Trans Commun, 2016, 64: 4983–4996
10 Zeng Y, Zhang R. Energy-efficient UAV communication with trajectory optimization. IEEE Trans Wireless Commun, 2017, 16: 3747–3760
11 Wu Q Q, Zhang R. Common throughput maximization in UAV-enabled OFDMA systems with delay consideration. IEEE Trans Commun, 2018, 66: 6614–6627
12 Zhan C, Zeng Y, Zhang R. Energy-efficient data collection in UAV enabled wireless sensor network. IEEE Wireless Commun Lett, 2018, 7: 328–331
13 Jeong S, Simeone O, Kang J. Mobile edge computing via a UAV-mounted cloudlet: optimization of bit allocation and path planning. IEEE Trans Veh Technol, 2018, 67: 2049–2063
14 Zhang X C, Zhang J, Xiong J, et al. Energy-efficient multi-UAV-enabled multiaccess edge computing incorporating NOMA. IEEE Internet Things J, 2020, 7: 5613–5627
15 Mozaffari M, Saad W, Bennis M, et al. Mobile unmanned aerial vehicles (UAVs) for energy-efficient Internet of Things communications. IEEE Trans Wireless Commun, 2017, 16: 7574–7589
16 Wu Q Q, Zeng Y, Zhang R. Joint trajectory and communication design for multi-UAV enabled wireless networks. IEEE Trans Wireless Commun, 2018, 17: 2109–2121
17 Shen C, Chang T H, Gong J, et al. Multi-UAV interference coordination via joint trajectory and power control. IEEE Trans Signal Process, 2020, 68: 843–858

18 Wang T H, Pang X W, Tang J, et al. Time and energy efficient data collection via UAV. Sci China Inf Sci, 2022, 65: 182302

19 Sheng M, Zhao C X, Liu J Y, et al. Energy-efficient trajectory planning and resource allocation in UAV communication networks under imperfect channel prediction. Sci China Inf Sci, 2022, 65: 222301

20 Tong Y Q, Sheng M, Liu J Y, et al. Energy-efficient UAV-NOMA aided wireless coverage with massive connections. Sci China Inf Sci, 2023, 66: 222303

21 Zhang T K, Chen C B, Xu Y, et al. Joint task scheduling and multi-UAV deployment for aerial computing in emergency communication networks. Sci China Inf Sci, 2023, 66: 192303

22 Lowe R, Wu Y, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, 2017. 6382–6393

23 Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518: 529–533

24 Zhou X H, Zhang X C, Zhao H T, et al. Constrained soft actor-critic for energy-aware trajectory design in UAV-aided IoT networks. IEEE Wireless Commun Lett, 2022, 11: 1414–1418

25 Liu C H, Chen Z Y, Tang J, et al. Energy-efficient UAV control for effective and fair communication coverage: a deep reinforcement learning approach. IEEE J Sel Areas Commun, 2018, 36: 2059–2070

26 Zhang R, Wang M, Cai L X, et al. Learning to be proactive: self-regulation of UAV based networks with UAV and user dynamics. IEEE Trans Wireless Commun, 2021, 20: 4406–4419

27 Yan C, Xiang X J, Wang C, et al. PASCAL: population-specific curriculum-based MADRL for collision-free flocking with large-scale fixed-wing UAV swarms. Aerospace Sci Tech, 2023, 133: 108091

28 Yan C, Wang C, Xiang X J, et al. Collision-avoiding flocking with multiple fixed-wing UAVs in obstacle-cluttered environments: a task-specific curriculum-based MADRL approach. IEEE Trans Neural Netw Learn Syst, 2023. doi: 10.1109/TNNLS.2023.3245124

29 Zhang W Q, Wang Q, Liu X, et al. Three-dimension trajectory design for multi-UAV wireless network with deep reinforcement learning. IEEE Trans Veh Technol, 2021, 70: 600–612

30 Cui J J, Liu Y W, Nallanathan A. Multi-agent reinforcement learning-based resource allocation for UAV networks. IEEE Trans Wireless Commun, 2020, 19: 729–743

31 Yuan Y X, Lei L, Vu T X, et al. Energy minimization in UAV-aided networks: actor-critic learning for constrained scheduling optimization. IEEE Trans Veh Technol, 2021, 70: 5028–5042

32 Zhong R K, Liu X, Liu Y W, et al. Multi-agent reinforcement learning in NOMA-aided UAV networks for cellular offloading. IEEE Trans Wireless Commun, 2022, 21: 1498–1512

33 Zhang X C, Zhao H T, Wei J B, et al. Cooperative trajectory design of multiple UAV base stations with heterogeneous graph neural networks. IEEE Trans Wireless Commun, 2023, 22: 1495–1509

34 Qin Z Q, Liu Z H, Han G J, et al. Distributed UAV-BSs trajectory optimization for user-level fair communication service with multi-agent deep reinforcement learning. IEEE Trans Veh Technol, 2021, 70: 12290–12301

35 Foerster J, Farquhar G, Afouras T, et al. Counterfactual multi-agent policy gradients. In: Proceedings of the 32nd Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, 2018. 2974–2982

36 Sunehag P, Lever G, Gruslys A, et al. Value-decomposition networks for cooperative multi-agent learning. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, 2018. 2085–2087

37 Rashid T, Samvelyan M, Schroeder C, et al. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In: Proceedings of the 35th International Conference on Machine Learning, 2018. 4295–4304

38 Yu C, Velu A, Vinitsky E, et al. The surprising effectiveness of PPO in cooperative, multi-agent games. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 35: 24611–24624

39 Yuan L, Wang J H, Zhang F X, et al. Multi-agent incentive communication via decentralized teammate modeling. In: Proceedings of Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, 2022. 9466–9474

40 Bettini M, Shankar A, Prorok A. System neural diversity: measuring behavioral heterogeneity in multi-agent learning. 2023. ArXiv:2305.02128

41 Bettini M, Shankar A, Prorok A. Heterogeneous multi-robot reinforcement learning. In: Proceedings of International Conference on Autonomous Agents and Multiagent Systems, 2023. 1485–1494

42 Marks B R, Wright G P. A general inner approximation algorithm for nonconvex mathematical programs. Oper Res, 1978, 26: 681–683

43 Xu Y Y, Yin W T. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM J Imag Sci, 2013, 6: 1758–1789

44 Ding R J, Gao F, Shen X S. 3D UAV trajectory design and frequency band allocation for energy-efficient and fair communication: a deep reinforcement learning approach. IEEE Trans Wireless Commun, 2020, 19: 7796–7809

45 Calvo J A, Dusparic I. Heterogeneous multi-agent deep reinforcement learning for traffic lights control. In: Proceedings of Conference on Artificial Intelligence and Cognitive Science, 2018. 2–13

46 Zheng S, Trott A, Srinivasa S, et al. The AI Economist: taxation policy design via two-level deep multiagent reinforcement learning. Sci Adv, 2022, 8: eabk2607

47 Jain R K, Chiu D-M W, Hawe W R. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. 1998. ArXiv:cs/9809099

48 Ha D, Dai A, Le Q V. HyperNetworks. In: Proceedings of International Conference on Learning Representations (ICLR), 2017

49 Sutton R S, Barto A G. Reinforcement Learning: An Introduction. Cambridge: The MIT Press, 2018

50 Goodfellow I, Bengio Y, Courville A. Deep Learning. Cambridge: The MIT Press, 2016

51 Chung J, Gulcehre C, Cho K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. In: Proceedings of NIPS 2014 Workshop on Deep Learning, 2014

52 Hu Y, Chen M Z, Saad W, et al. Distributed multi-agent meta learning for trajectory design in wireless drone networks. IEEE J Sel Areas Commun, 2021, 39: 3177–3192

53 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, 2017

54 Zaheer M, Kottur S, Ravanbakhsh S, et al. Deep sets. In: Proceedings of Advances in Neural Information Processing Systems, 2017. 3391–3401

55 Battaglia P W, Hamrick J B, Bapst V, et al. Relational inductive biases, deep learning, and graph networks. 2018. ArXiv:1806.01261