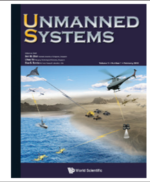
Unmanned Systems, Vol. 14, No. 2 (2026) 523–543

DOI: 10.1142/S2301385026410013

UNMANNED SYSTEMS
https://www.worldscientific.com/worldscinet/us



Enhanced Air–Ground Coordination with the HUCC-MAPPO Algorithm for Improved Multi-UAV Reconnaissance Efficiency

Zhengyang Cao [*,†], Gang Chen [*,‡]

[*] *State Key Laboratory of Strength and Vibration for Mechanic Structures, School of Aerospace Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, P. R. China*

[†] *Xi'an ASN UAV Technology Co., Ltd., Xi'an, Shaanxi 710065, P. R. China*

A novel air–ground coordination (AGC) system is developed, integrating multi-agent proximal policy optimization (MAPPO) with Voronoi diagram-based path planning to revolutionize task allocation and execution in complex environments. This advanced approach significantly enhances the efficiency and precision of coordination between unmanned aerial vehicles (UAVs) and ground operators, ensuring superior mission performance. Key improvements of the proposed system include adaptive task allocation and Voronoi diagram-based path planning, which effectively optimize task area coverage and mission execution. In simulations, the HUCC-MAPPO algorithm demonstrates exceptional improvements in simulation, surpassing conventional methods in task coverage, mission efficiency, and obstacle avoidance. Specific experimental results show significant enhancements in task area coverage by 15%, reduction in path lengths by 12% and an increase in task completion rates by 18% compared to baseline methods. These results highlight its strong potential for real-world applications, particularly in multi-UAV reconnaissance missions where precise and dynamic coordination is crucial.

Keywords: Air–ground coordination system; multi-UAV; task allocation; path planning; MAPPO; Voronoi diagram.

1. Introduction

In contemporary reconnaissance operations, the intricate terrain and the ever-changing nature of targets pose significant challenges to the effective acquisition, processing and transmission of positional data [1]. Unmanned aerial vehicle (UAV) systems, renowned for their high mobility and flexible deployment, have emerged as indispensable tools in addressing these challenges [2–4]. In real-world environments, air–ground coordination (AGC) systems face dynamic challenges, such as task prioritization, obstacle avoidance and power management. Through coordinated control, AGC systems significantly improve task coverage and efficiency, demonstrating potential in applications like large-scale reconnaissance, emergency rescue and disaster response [5–7]. Despite the significant potential of AGC systems in large-area reconnaissance, emergency rescues and disaster response, achieving efficient coordination within these systems in complex operational environments presents several technical challenges. These challenges include the need for adaptive task allocation to handle dynamic priority changes, reliable path planning to ensure safe navigation through complex terrains, and efficient integration of multi-source information to maintain high-coordination efficiency [8–11]. During task execution, UAVs must dynamically plan their trajectories to avoid collisions and optimize operational efficiency. Additionally, efficient coordination between UAVs and ground systems, ensuring the rapid transfer and processing of information, is crucial for enhancing overall task completion effectiveness. AGC systems face multiple challenges due to dynamic and complex factors in real-world environments. Based on the criteria of practicality, quantifiability, controllability, and adaptability, dynamic task priority adjustment, real-time obstacle avoidance, dynamic power management,

Received 26 August 2024; Accepted 27 January 2025; Published 17 June 2025. This paper was recommended for publication in its revised form by Special Issue Editors, Hai-Tao Zhang, Chen Lyu and Bin-Bin Hu.
Email Address: ‡*aachengang@xjtu.edu.cn*
‡ Corresponding author.

multi-objective path optimization and multi-UAV coordination have been selected. These factors significantly impact system performance, ensuring efficient adaptability and operational performance in dynamic environments.

With the rapid advancement of artificial intelligence (AI) technologies, particularly in reinforcement learning (RL) [12] and deep learning (DL) [13], the application of these cutting-edge technologies to augment the intelligence of the AGC system has become a central focus of contemporary research. DL excels at efficiently processing and analyzing complex datasets, while RL iteratively refines decision-making processes in dynamic and uncertain environments [14–16]. This study leverages deep reinforcement learning (DRL) combined with Voronoi diagram-based path planning to develop and validate a coordination control strategy for UAVs and ground operators. This strategy aims to substantially enhance task allocation, path optimization, and the overall efficiency of human–UAV coordination in complex environments, thereby markedly improving the task execution efficiency and reliability of the AGC system in real-world applications.

Building on these advancements, this study specifically aims to design an intelligent task allocation algorithm and path planning strategy to optimize the execution processes of the AGC system, thereby enhancing the efficiency and precision of task completion. By employing online learning and a DRL method, the multi-UAV system can adaptively respond to dynamic and unpredictable environments, enhancing its robustness and flexibility. Additionally, by leveraging multi-source information acquired and integrated by UAVs, an intelligent decision-making algorithm is employed to improve coordination efficiency between UAVs and ground operators. This approach ensures the rapid transfer and processing of information, ultimately optimizing task execution outcomes. Key innovations of this study include a flexible task allocation mechanism that adapts to dynamic priority changes, a Voronoi-based path planning algorithm that ensures precise navigation in complex terrains and an integrated multi-source information processing strategy that enhances coordination efficiency in AGC systems. The schematic diagram of the AGC system is illustrated in Fig. 1. The main contributions and features of this study are as follows:

- Adaptive Task Allocation Mechanism: Traditional AGC systems generally use static priority or predefined allocation methods that lack flexibility for complex, multi-task scenarios. In contrast, the multi-agent proximal policy optimization (MAPPO)-based task allocation framework employs a centralized training and decentralized execution strategy, enhancing coordination between UAVs and ground operators. By dynamically adjusting task loads based on task priorities and geographical conditions, this approach addresses imbalances in resource distribution and significantly improves task completion efficiency and responsiveness.

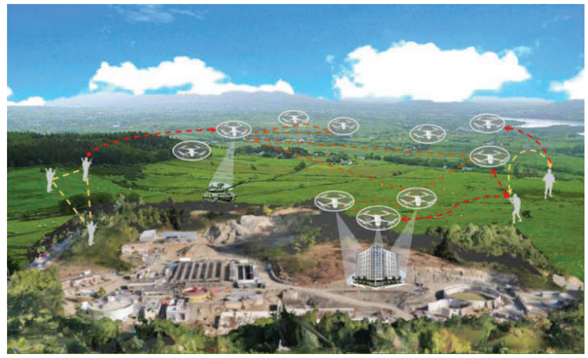


Fig. 1. Schematic diagram of the AGC system.

- Voronoi-Based Path Planning Algorithm: In obstacle-dense environments, traditional AGC path planning methods often lack the efficiency required for optimal route selection and safety. The Voronoi-based path planning algorithm dynamically optimizes paths between UAVs and ground operators, enabling effective obstacle navigation, reducing computation time and enhancing operational reliability. This approach significantly improves both path optimization and the system's obstacle avoidance capabilities in complex environments.
- HUCC-MAPPO Coordination Control: Traditional AGC systems for human–machine coordination often rely on static communication protocols, limiting the ability to perform effective real-time coordination and decision-making. The HUCC-MAPPO algorithm utilizes multi-source information integration to optimize human–machine collaboration by enabling adaptive adjustments to coordination parameters. This enhances mission coverage, optimizes resource allocation and ensures high mission efficiency, particularly in complex and dynamic environments.

The remainder of this paper is structured as follows: Section 2 provides a review of existing work on task allocation, path planning and human–machine coordination control, emphasizing current research gaps and suggesting future research directions. Section 3 introduces the system model and problem formulation, elaborating on the key components and mathematical models. Section 4 offers a comprehensive overview of the HUCC-MAPPO algorithm, outlining its design and implementation from an algorithmic perspective. Section 5 describes the experimental design and evaluation metrics, evaluating the effectiveness of the HUCC-MAPPO algorithm in comparison to other methods. Section 6 summarizes the research findings and outlines potential future research directions.

2. Related Work

In the field of AGC system and human–machine coordination, numerous studies have explored critical issues such as task allocation, path planning and coordination control. Recent research achievements reveal significant progress while also pointing out existing shortcomings. Identifying these advancements and gaps is crucial for developing more effective HUCC method in future research.

2.1. *Task allocation*

Efficient task allocation is pivotal in enhancing task performance within the AGC system. Although traditional task allocation methods, such as ant colony optimization, social network-assisted optimization, and mixed integer linear programming, have been explored in various contexts, their application to the AGC system presents unique challenges. For example, Liu and Zhao [17] utilized an acyclic directed graph to address the role-oriented task allocation problem, applying ant colony optimization to derive optimized solutions. While this approach shows potential, it is hindered by high-computational complexity and slow convergence, which are particularly problematic in multi-UAV systems and complex task environments typical of the AGC system. Similarly, Gao *et al.* [18] developed a social network-assisted mobile user recruitment method that proved effective in multi-UAV task allocation scenarios. However, its limitations become evident in highly dynamic environments and complex task distributions, which are common in the AGC system. Furthermore, Ma *et al.* [19] explored a mixed integer linear programming algorithm that integrates UAVs and unmanned ground vehicles, significantly enhancing task execution efficiency and coordination. Despite these advancements, challenges remain in prioritizing tasks and managing resource constraints, highlighting the need for more adaptable and robust solutions within the AGC system.

While traditional task allocation methods have improved task execution efficiency across various applications, they encounter significant challenges in the context of the AGC system, particularly under dynamic conditions and large-scale task management scenarios. These methods often struggle with computational complexity and slow convergence, which limits their effectiveness in rapidly changing environments. This study introduces the HUCC-MAPPO algorithm, which addresses these challenges by enhancing coordination efficiency and system responsiveness. Specifically, the HUCC-MAPPO algorithm integrates DRL with multi-agent coordination strategies, allowing for dynamic task allocation and continuous adaptation to real-time environmental changes. By optimizing decision-making processes and adjusting task distribution based on current data, the algorithm improves the division and coverage of task areas, effectively manages dynamic changes, and enhances overall system efficiency in AGC system operations.

2.2. *Path planning*

Path planning is a critical component of task execution within the AGC system. Traditional path planning methods, such as the Dijkstra algorithm, A* algorithm and various graph-based techniques, have proven effective in specific scenarios. However, these methods face significant challenges when coordinating path planning for multi-UAV and ground systems, particularly in ensuring path conflict avoidance and achieving global optimality. For example, Liu *et al.* [20] applied the Dijkstra algorithm for global path planning in smart vehicles, while Lai *et al.* [21] introduced a center-constrained weighted A* algorithm for robot path planning on two-dimensional grid maps. Although effective, these methods are limited in their applicability to dynamic and complex environments.

To overcome these limitations, Vashisth *et al.* [22] employed DRL with dynamic graphs for adaptive and informative path planning. This approach significantly reduced path conflicts and resource consumption, demonstrating its potential in dynamic environments. Similarly, Venkatasivarambabu *et al.* [23] proposed a path planning method that combines dynamic programming with probabilistic routing, effectively enhancing UAV navigation capabilities in uncertain environments. Additionally, Cui *et al.* [24] utilized a DRL algorithm to enhance UAV navigation performance through large-scale multiple-input multiple-output (MIMO) technology.

Despite these advancements, the reliability and safety of path planning in complex terrains and dynamic environments still require further improvement. To address these challenges, this study proposes using Voronoi diagram for path planning, integrating the movement paths of ground operators. This method minimizes the distance to task points, thereby improving task execution efficiency. It significantly enhances the precision and real-time performance of path planning, ensuring effective navigation for both UAVs and ground systems in complex environments.

2.3. *Human–machine coordination control*

Effective human–machine coordination control is essential for the seamless operation of the AGC system. Traditional human–machine coordination primarily relies on predefined rules and protocols, often exhibiting limited flexibility and adaptability. With the advancement of AI technologies, researchers have increasingly explored intelligent algorithm-based coordination methods. For instance, Lei *et al.* [25] achieved dynamic coordination and task allocation between underwater vehicles and ground operators using DL and RL, significantly enhancing task completion rates and the reliability of information transmission. Similarly, Haesevoets *et al.* [26] investigated the willingness of individuals to cooperate with machines, identifying varying responses based on differing levels of human versus machine involvement. Additionally, Mazhar *et al.* [27] developed a framework for safe, real-time human–robot coordination using static hand gestures and 3D skeleton extraction, integrating it into the OpenPHRI library, which facilitates physical human–robot interactions.

Despite these advancements, current research in human–machine coordination control frequently focuses on static human–robot interaction, lacking sufficient exploration of dynamic and real-time coordination between humans and UAVs. Most existing methods emphasize understanding minimal human actions, without adequately addressing the complexities of dynamic human–UAV interactions. This study aims to bridge these gaps by constructing an AGC system that enhances information exchange and task coordination efficiency between UAVs and ground operators. The system implements a coordination control algorithm based on MAPPO, ensuring the efficient execution of coordination tasks. This algorithm enhances the flexibility and adaptability of the AGC system in complex and dynamic environments, enabling more sophisticated and effective human–UAV interactions.

2.4. *Research shortcomings and prospects*

Existing research in task allocation, path planning and human–machine coordination within the AGC system has made significant strides. However, several challenges persist, particularly in achieving high levels of real-time performance, safety, reliability, large-scale task management and global optimality in complex and dynamic environments. Traditional algorithms often lack the capacity for real-time adjustments, as they rely on static protocols and predefined paths, which limits their effectiveness in rapidly changing environments. This inflexibility poses significant challenges for maintaining safe navigation and effective collision avoidance in intricate terrains, as many systems fail to robustly manage real-time environmental changes.

Moreover, existing systems frequently struggle to maintain consistent task execution rates under fluctuating conditions, particularly when integrating operators for real-time decision-making. The complexity of managing and efficiently allocating resources in large-scale, multi-UAV scenarios often results in suboptimal task area coverage and resource utilization [28]. Furthermore, achieving global optimality in path planning, which involves minimizing path length and enhancing path efficiency, remains a formidable challenge in multi-agent system operating within dynamic contexts [29–31].

To address these challenges, this study proposes a comprehensive HUCC methodology based on the MAPPO algorithm. This methodology leverages DRL to enhance the efficiency and accuracy of task allocation, improve the precision and real-time performance of path planning and optimize coordination control. By integrating Voronoi diagram for optimized path planning and utilizing multi-source sensor data, the system significantly enhances the adaptability and precision of UAV operations in dynamic environments. This approach not only mitigates the limitations of traditional algorithms by enabling real-time adjustments and robust decision-making but also lays the groundwork for advancing an intelligent, adaptable and robust AGC system capable of achieving global optimality and effective resource utilization in complex scenarios.

3. System Model and Problem Description

3.1. *System components*

This study presents a comprehensive HUCC methodology within the AGC system, designed to effectively integrate the operations of UAVs and ground operators for efficient execution of complex real-time tasks. The AGC system is structured around three core components, each playing a critical role in the system's functionality: the task allocation module, the path planning module and the human–UAV coordination control module, as depicted in Fig. 2.

(1) Task allocation module
 This module is tasked with dynamically assigning tasks to UAVs and ground operators, based on real-time environmental data and mission requirements. It employs

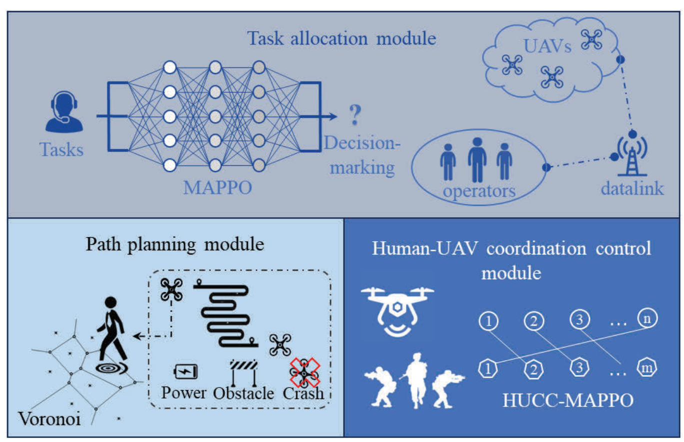


Fig. 2. Schematic diagram of the AGC system components.

an advanced algorithm, augmented by the MAPPO framework, to enhance the precision and real-time performance of task distribution. By incorporating multi-source sensor data, this module optimizes resource utilization and improves task completion rates, ensuring that resources are allocated efficiently and mission objectives are met.

(2) Path planning module
This module is responsible for generating optimal paths for UAVs and ground operators. It utilizes Voronoi diagram and multi-objective optimization techniques to address challenges such as obstacle avoidance, energy efficiency and crash prevention. By ensuring that paths are both safe and efficient, this module minimizes the distance traveled and the time required for task completion, thereby enhancing overall operational efficiency.

(3) Human–UAV coordination control module
This module facilitates seamless coordination between UAVs and ground operators, ensuring effective interaction throughout mission execution. It leverages the HUCC-MAPPO algorithm to optimize decision-making processes by integrating multi-source information. This approach enhances real-time performance and robustness, enabling efficient information exchange and coordinated task execution, which are critical for maintaining system reliability and mission success.

3.2. *Problem description*

The AGC system, designed for multi-UAV operations, encounters significant challenges in performing coordinated reconnaissance within complex and dynamic target areas. These challenges include effectively allocating tasks, planning optimal paths and ensuring seamless coordination between UAVs and ground operators in real time. The HUCC-MAPPO algorithm, proposed in this study, addresses these challenges by providing an integrated approach that enhances the AGC system's capability to perform coordinated reconnaissance tasks efficiently and reliably.

(1) Addressing task allocation challenges
Task allocation is enhanced by dynamically adapting to each UAV's performance, capacity and the specific demands of the mission. Through continuous adjustment based on real-time data, this approach maximizes system efficiency and responsiveness. Prioritizing tasks and managing dependencies ensures that the AGC system achieves comprehensive coverage of the target area.

(2) Optimizing path planning
Path planning is optimized by integrating Voronoi diagrams and multi-objective optimization to generate safe and efficient routes. This method reduces unnecessary overlaps, minimizes travel distance and ensures effective obstacle avoidance, thereby improving both the operational efficiency and safety of the AGC system during reconnaissance missions.

(3) Enhancing coordination and execution
Coordination between UAVs and ground operators is streamlined by integrating multi-source information and enabling adaptive decision-making. This real-time synchronization facilitates dynamic task execution, resource reallocation and quick adjustments to changing conditions, ensuring that the AGC system maintains high levels of performance and reliability in complex environments.

By addressing the core challenges of task allocation, path planning and coordination, the proposed approach significantly enhances the AGC system's effectiveness in coordinated reconnaissance. This methodology ensures greater efficiency, safety and reliability, enabling precise and effective mission execution in complex scenarios.

3.3. *Mathematical model*

In complex and dynamic environments, effective coordination between UAVs and ground operators requires a well-structured mathematical framework. The Voronoi diagram serves as a foundational approach in this framework, enabling precise spatial partitioning that is crucial for optimizing task allocation, path planning and coordination. This spatial approach ensures efficient resource distribution, minimizes path conflicts and allows for dynamic adjustments based on real-time data. To formalize these processes, this study develops mathematical models that define the optimization objectives and constraints, addressing the challenges of adapting to changing task demands and enhancing coordination within the AGC system.

3.3.1. *Construction of Voronoi diagram*

The Voronoi diagram is a fundamental tool in computational geometry, offering an efficient method for partitioning a given space based on the proximity of a set of points. In the context of the AGC system, the Voronoi diagram is employed to optimally divide the task area, ensuring that each task point is assigned to the most appropriate UAV and ground operator team. This spatial partitioning facilitates efficient task allocation and path planning, thereby enhancing the overall efficiency and coordination of the AGC system.

The task area is defined as a three-dimensional region $D \subseteq \mathbb{R}^3$, encompassing M UAVs and N ground operators. Let $P = \{p_1, p_2, \ldots, p_{M+N}\}$ represent the set of all UAVs and ground operator positions, where each point p_v corresponds to the position of a UAV or ground operator in the task area. The Voronoi diagram partitions this area into K cells, where K is the number of task points, with each cell representing the region closest to a particular task point.

Mathematically, the Voronoi cell associated with a point $p_v \in P$ is defined as follows:

$$V(p_v) = \{q \in D | d(p_v, q) \leq d(p_w, q), \quad \forall v \neq w\}, \tag{1}$$

where $d(p_v, q)$ denotes the Euclidean distance between points p_v and q.

The construction of the Voronoi diagram involves the following steps:

(1) Initialization: Identify the positions of all UAVs and ground operators within the task area.
(2) Distance calculation: For each point in the task area, calculate the Euclidean distance to UAVs and ground operators.
(3) Cell assignment: Assign each point to the closest UAV and ground operator team, thereby forming the Voronoi cells.

The Voronoi diagram partitions the task area into sub-areas based on the proximity to task points, with each sub-area associated with the nearest UAV and ground operator. As illustrated in Fig. 3, each cell is defined by its proximity to a task point, ensuring that the task area is divided efficiently. Each task point within a Voronoi cell serves as the focal point for task allocation, effectively preventing both overlap and underutilization of resources.

To calculate the path length from a UAV or ground operator to a task point, the total distance traveled through the Voronoi cells can be expressed as follows:

$$d_{ik} = \sqrt{(x_{u_i} - x_{t_k})^2 + (y_{u_i} - y_{t_k})^2 + (z_{u_i} - z_{t_k})^2}, \tag{2}$$

$$d_{jk} = \sqrt{(x_{g_j} - x_{t_k})^2 + (y_{g_j} - y_{t_k})^2 + (z_{g_j} - z_{t_k})^2}, \tag{3}$$

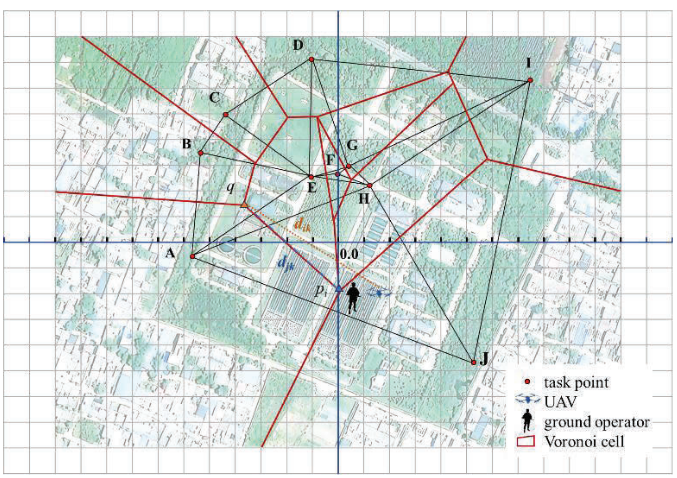


Fig. 3. Schematic diagram of Voronoi diagram construction.

where $u_i = (x_{u_i}, y_{u_i}, z_{u_i})$ represents the position of the ith UAV, $t_k = (x_{t_k}, y_{t_k}, z_{t_k})$ represents the position of the kth task point, and $g_j = (x_{g_j}, y_{g_j}, z_{g_j})$ represents the position of the jth ground operator.

Once the distances are calculated, the Voronoi diagram is used to efficiently partition the task area among UAVs and ground operators. Each task point serves as the center of a Voronoi cell, and the nearest UAV and ground operator team is assigned to cover the task point and its surrounding area. This partitioning minimizes path lengths, prevents overlapping responsibilities and maximizes resource utilization, thereby enhancing the overall efficiency and effectiveness of the AGC system in complex and dynamic environments.

3.3.2. *Optimization objectives and constraints*

To achieve efficient task allocation and path planning in the AGC system, optimization objectives and constraints are integrated into a unified model. This model addresses the dynamic and complex nature of the operational environment, ensuring efficient coordination and resource utilization, thereby improving overall system performance and reliability. Assume $T = \{t_1, t_2, \ldots, t_k\}$ represents the task set. For UAV u_i and ground operator g_j, each task t_k has a specific location and required completion time. Task allocation optimization aims to minimize the total task execution time, defined as the time to reach the task point, while ensuring efficient coordination between UAVs and ground operators. The mathematical expression is as follows:

$$\min\left(\sum_{i=1}^{M}\sum_{k=1}^{K} a_{ik} \cdot \tau_{ik} + \sum_{j=1}^{N}\sum_{k=1}^{K} a_{jk} \cdot \tau_{jk}\right), \tag{4}$$

where a_{ik} and a_{jk} are elements of the task allocation matrix, and τ_{ik}, τ_{jk} represent the time required for UAV u_i and ground operator g_j to execute task t_k, respectively.

The task allocation variables a_{ik} and a_{jk} are defined as follows:

$$a_{ik} = \begin{cases} 1 & \text{if task } t_k \text{ is assigned to UAV } u_i, \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

$$a_{jk} = \begin{cases} 1 & \text{if task } t_k \text{ is assigned to ground operator } g_j, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

For calculating the time τ_{ik} and τ_{jk}, the following expressions are used:

$$\tau_{ik} = \frac{d_{ik}}{v_{u_i}} \quad \text{and} \quad \tau_{jk} = \frac{d_{jk}}{v_{g_j}}, \tag{7}$$

where d_{ik} and d_{jk} are the distances calculated using the Euclidean distance formula, and v_{u_i} and v_{g_j} are the speeds of the UAV and ground operator, respectively.

To correspond with the Voronoi diagram approach and ensure collision-free coordination among multiple UAVs, the allocation and path constraints can be expressed as follows:

$$V(p_v) \cap V(p_w) = \emptyset, \quad \forall\, v \neq w. \tag{8}$$

This constraint ensures that each task point t_k is assigned to the UAV and ground operator team responsible for the Voronoi cell it resides in, while also maintaining collision-free path planning. By structuring task allocation within distinct Voronoi cells, the system minimizes path conflicts and optimizes resource distribution in dynamic environments.

Task allocation constraints ensure that each task point is assigned to a team of two UAVs and one ground operator, with the following constraints:

$$\sum_{i=1}^{M} a_{ik} + \sum_{j=1}^{N} a_{jk} = 1, \quad \forall\, k, \tag{9}$$

$$\sum_{i=1}^{M} a_{ik} \leq \text{max\_load}_u, \quad \forall\, i, \tag{10}$$

$$\sum_{j=1}^{N} a_{jk} \leq \text{max\_load}_g, \quad \forall\, j. \tag{11}$$

Path planning optimization seeks to minimize the path lengths from the current positions of UAVs and ground operators to the task locations while avoiding collisions. The mathematical expression is as follows:

$$\min\left(\sum_{i=1}^{M}\sum_{k=1}^{K} d_{ik} + \sum_{j=1}^{N}\sum_{k=1}^{K} d_{jk}\right). \tag{12}$$

To ensure that the number of tasks assigned to UAVs and ground operators does not exceed their maximum capacity, and to safely avoid path overlaps and conflicts, the following constraints are established. Each task t_k can only be assigned to two UAVs and one ground operator, ensuring effective task coverage without resource overload, as shown in Fig. 4.

Path conflict constraints are defined as follows:

$$\text{path}(u_i, t_k) \cap \text{path}(u_m, t_k) = \emptyset, \quad \forall\, i \neq m, \tag{13}$$

$$\text{path}(g_j, t_k) \cap \text{path}(g_n, t_k) = \emptyset, \quad \forall\, j \neq n. \tag{14}$$

In path planning, UAVs are assumed to maintain a constant altitude, so the paths, $\text{path}(u_i, t_k)$ and $\text{path}(g_j, t_k)$ are calculated primarily in the horizontal plane, considering only the x and y distances:

$$\text{path}(u_i, t_k) = \sqrt{(x_{u_i} - x_{t_k})^2 + (y_{u_i} - y_{t_k})^2}, \tag{15}$$

$$\text{path}(g_j, t_k) = \sqrt{(x_{g_j} - x_{t_k})^2 + (y_{g_j} - y_{t_k})^2}. \tag{16}$$

The calculation process for other UAVs or ground operators, such as $\text{path}(u_m, t_k)$ and $\text{path}(g_n, t_k)$, follows the same approach, focusing solely on horizontal distances. These optimization objectives and constraints ensure the efficiency and safety of task allocation and path planning in the AGC system. This comprehensive approach integrates task allocation and path planning into a unified model, specifically designed to address the dynamic and complex nature of tasks and environments. By focusing on horizontal path calculations and avoiding path conflicts, the model enhances overall system performance and reliability. Implementing this unified optimization model enables the AGC system to dynamically adapt to real-time changes in task demands and environmental conditions [32–34].

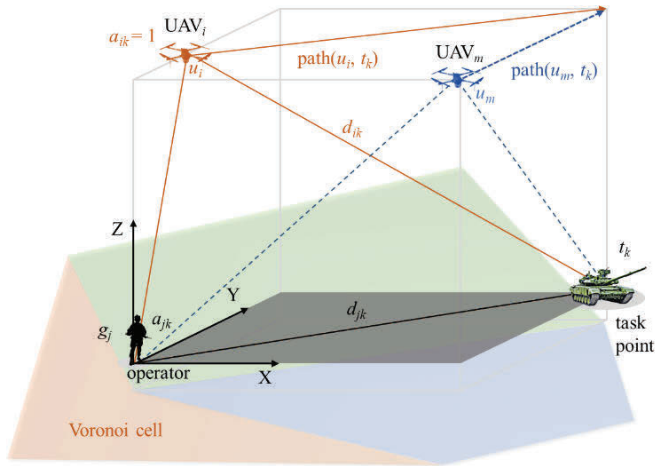


Fig. 4. Schematic diagram of optimization objectives and constraints in the AGC system.

3.3.3. *Task allocation*

In dynamic and complex environments, task demands and environmental conditions can change rapidly [35]. Efficient and intelligent task allocation to UAVs and ground operators is crucial for effective coordination. Task priority is a key factor, considering the varying levels of importance and urgency of different tasks, which necessitates dynamic adjustments in task allocation strategy. The priority of each task t_k is calculated as follows:

$$\text{Pri}(t_k) = \alpha \cdot U(t_k) + \alpha' \cdot E(t_k), \tag{17}$$

where α and α' are weight coefficients, ranging from 0 to 1, reflecting the relative importance of the task's nature and urgency. $U(t_k)$ represents the task's criticality or impact, while $E(t_k)$ accounts for time sensitivity or deadlines. The task priority function $\text{Pri}(t_k)$ enables the system to prioritize tasks dynamically, balancing criticality and urgency to adaptively meet real-time demands. Given the critical nature of deadlines, accounts for time sensitivity or deadlines, $E(t_k)$ ensuring that tasks are prioritized based on their urgency and the time available for their completion.

Task allocation must also account for dependencies, where some tasks need to be completed before others can begin. By analyzing flight data, dependencies between tasks can be identified, enabling optimized task allocation strategy. The dependency between tasks t_k and t_l is defined as follows:

$$D_{kl} = \begin{cases} 1 & \text{if task } t_l \text{ depends on task } t_k, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

Balanced load distribution among UAVs and ground operators is crucial to avoid the overuse or idleness of resources. Factors such as UAV positions, velocities, attitudes and ground operators' locations relative to the centroid of the Voronoi cell are considered. The resource utilization model ensures efficient coordination by optimizing the use of each resource:

$$R(u_i) = \frac{\sum_{k=1}^{K} D_{ik} \cdot r_{ik}}{\text{max\_load}_u}, \tag{19}$$

$$R(g_j) = \frac{\sum_{k=1}^{K} D_{jk} \cdot r_{jk}}{\text{max\_load}_g}, \tag{20}$$

where $R(u_i)$ and $R(g_j)$ represent the resource utilization rates of UAV u_i and ground operator g_j, respectively. r_{ik} and r_{jk} represent the resource requirements for task t_k. Specifically, r_{ik} refers to the energy consumption of UAV u_i in kilowatt-hours (kWh), and r_{jk} refers to the energy expenditure of ground operator g_j, also converted to kWh.

To ensure efficient and reasonable task allocation to UAVs and ground operators, the objective function for task allocation is expressed as follows:

$$\min \left(\begin{array}{c} \sum_{i=1}^{M} \sum_{k=1}^{K} D_{ik} \cdot (\beta_1 \cdot \text{Pri}(t_k) + \beta_2 \cdot R(u_i)) \\ + \\ \sum_{i=1}^{N} \sum_{k=1}^{K} D_{jk} \cdot (\beta_1 \cdot \text{Pri}(t_k) + \beta_2 \cdot R(g_j)) \end{array} \right), \tag{21}$$

where β_1 and β_2 are weight coefficients representing the importance of task priority and resource utilization in the objective function.

To complete the task allocation process, the system dynamically adjusts the allocation of tasks based on real-time data and changing environmental conditions. This involves continuously monitoring task progress, UAV and ground operator status and mission requirements. The goal is to ensure that tasks are completed efficiently and effectively. The system focuses on optimizing key aspects of performance, such as ensuring comprehensive mission coverage and maintaining high operational efficiency. By refining these performance elements, the system seeks to maximize task execution success while minimizing the required time and computational resources.

3.3.4. *Path planning*

In the AGC system involving UAVs and ground operators, path planning is crucial for ensuring efficient and safe task execution. To optimize performance, a path planning method utilizing the Voronoi diagram is employed. As defined in Eq. (1), the Voronoi diagram partitions the task area into cells, with each cell representing the region closest to a specific task point. This partitioning enables UAVs and ground operators to effectively navigate the environment by identifying and avoiding known obstacles and threats, thereby enhancing the overall efficiency and safety of the path planning process.

Minimizing path lengths is essential to enhancing task execution efficiency. Path planning should reduce the movement distance between UAVs, ground operators and task points, thereby conserving time and energy. This can be achieved by constructing a path planning model based on the Voronoi diagram. The path lengths for UAV u_i and ground operator g_j to the task points are calculated as d_{ik} and d_{jk} respectively, as defined in Eqs. (2) and (3). The distance between UAVs and ground operators, d_{ij} represents their coordination distance and is calculated as follows:

$$d_{ij} = \sqrt{(x_{u_i} - x_{g_j})^2 + (y_{u_i} - y_{g_j})^2 + (z_{u_i} - z_{g_j})^2}. \tag{22}$$

The path minimization objective function can be expressed as follows:

$$\min \left(\sum_{i=1}^{M} \sum_{k=1}^{K} d_{ik} + \sum_{j=1}^{N} \sum_{k=1}^{K} d_{jk} + \sum_{i=1}^{M} \sum_{k=1}^{N} d_{ij} \right). \tag{23}$$

To maintain effective communication between UAVs and ground operators, the following constraint must be observed:

$$d_{ij} \le R_{\text{datalink}}, \tag{24}$$

where R_{datalink} is the maximum distance for real-time control and information transmission between UAVs and ground operators [36].

In complex environments with changing obstacles, Voronoi path planning dynamically adjusts UAV paths to maintain safe distances while achieving optimal path selection. Obstacle avoidance is a key aspect of the path planning process, ensuring operational safety and reliability. To prevent UAVs and ground operators from encountering obstacles, the system must effectively detect and avoid them during mission execution. This involves real-time monitoring of the environment and dynamic adjustment of paths to avoid potential collisions. The path planning model includes a collision avoidance mechanism that accounts for dynamic changes and the diversity of obstacles, ensuring the validity and safety of the planned paths. The constraint for obstacle avoidance is defined as follows:

$$d(u_i, \text{obstacle}) \geq d_{\text{safe}} + r_{\text{turn}}, \tag{25}$$

where $d(u_i, \text{obstacle})$ is the actual distance between UAV u_i and the obstacle. d_{safe} is the safe distance, which is the minimum allowable distance between UAV and the obstacle to prevent collisions or dangerous proximity. r_{turn} is the minimum turning radius of the UAV, taking into account the UAV's dynamic characteristics and turning capabilities.

For ground operators, while there is no consideration of flight altitude, it remains crucial to maintain a safe distance from obstacles. Similar to UAVs, a minimum safe distance must be maintained:

$$d(g_j, \text{obstacle}) \geq d_{\text{safe\_ground}}, \tag{26}$$

where $d(g_j, \text{obstacle})$ is the actual distance between ground operator g_j and the obstacle. $d_{\text{safe\_ground}}$ is the minimum safe distance between the ground operator and the obstacle. This distance is determined by factors such as the operator's reaction time, the physical size of the equipment, and their movement speed.

Multi-objective optimization is necessary to balance various factors such as path length and time [37]. A multi-objective optimization model can help identify the optimal path planning solution by considering different optimization goals. The objective function for multi-objective optimization can be expressed as follows:

$$\min\left(\sum_{i=1}^{M}\sum_{k=1}^{K}(a_{ik}\cdot\hat{d}_{ik}+b_{ik}\cdot\hat{\tau}_{ik})+\sum_{j=1}^{N}\sum_{k=1}^{K}\times(a_{jk}\cdot\hat{d}_{jk}+b_{jk}\cdot\hat{\tau}_{jk})\right), \tag{27}$$

where $\hat{d}_{ik}$ and $\hat{d}_{jk}$ are the normalized path lengths [38], calculated as $\hat{d}_{ik} = \frac{d_{ik}}{\max(d)}$, $\hat{d}_{jk} = \frac{d_{jk}}{\max(d)}$, where $\max(d)$ is the maximum path length among all $\hat{d}_{ik}$ and $\hat{d}_{jk}$. $\hat{\tau}_{ik}$ and $\hat{\tau}_{jk}$ are the normalized task execution times, calculated as $\hat{\tau}_{ik} = \frac{\tau_{ik}}{\max(\tau)}$, $\hat{\tau}_{jk} = \frac{\tau_{jk}}{\max(\tau)}$, where $\max(\tau)$ is the maximum task execution time among all $\hat{\tau}_{ik}$ and $\hat{\tau}_{jk}$. a_{ik}, a_{jk}, b_{ik} and b_{jk} are weight coefficients for balancing the importance of path length and task execution time in the optimization goals.

To enhance operational sustainability, energy consumption is integrated as a key factor in the multi-objective optimization model. The objective function aims to balance path length, task execution time, and energy consumption, as shown below:

$$\min\left(\sum_{i=1}^{M}\sum_{k=1}^{K}d_{ik}+\beta_{\text{time}}\cdot T_{\text{execution}}+\gamma_{\text{energy}}\cdot E_{\text{energy}}\right), \tag{28}$$

where d_{ik} represents the distance between UAVs or ground operators and their assigned tasks. $T_{\text{execution}}$ indicates the time required to complete tasks. E_{energy} represents the total energy consumption, influenced by path length, UAV speed and payload. The coefficients β_{time} and γ_{energy} determine the relative importance of each term, allowing the model to adjust priorities based on mission requirements.

Energy consumption primarily depends on propulsion energy, which is directly affected by the UAV's velocity and the total distance traveled. In the optimization approach presented here, minimizing path lengths inherently leads to reduced energy usage, thereby enhancing operational sustainability for extended missions.

Path planning in the AGC system is essential for ensuring safe, efficient and sustainable task execution. By minimizing total flight distance, balancing task execution time and managing energy consumption, the system optimizes resource utilization and enhances operational safety. The model's ability to minimize both flight distance and energy usage reflects the system's efficiency in supporting extended missions, while effective obstacle avoidance ensures reliable navigation through complex terrains. Together, these factors, along with computational efficiency from the task allocation model, provide a comprehensive assessment of the AGC system's performance and adaptability in dynamic environments.

3.3.5. *Human–UAV coordination control*

In complex and dynamic environments, the effective coordination between UAVs and ground operators is essential for achieving comprehensive task area coverage and ensuring efficient task execution. The AGC system employs the Voronoi diagram to dynamically allocate task areas, optimizing coverage and minimizing resource wastage by leveraging real-time data.

During missions, UAVs and ground operators must synchronize their actions to maximize task efficiency. This involves continuously adjusting paths and reallocating tasks based on real-time information about the environment and task progress. The system integrates data from multiple sensors, including terrain features, obstacle locations and the real-time positions of UAVs and ground operators, to optimize coordination and task allocation effectively.

The dynamic task reallocation mechanism ensures that when changes occur in the task environment, tasks can be reassigned promptly to optimize resource utilization and mission efficiency. The mathematical formulation for task reallocation, represented as $\text{tra}(u_i, g_j, t_k)$, is as follows:

$$\text{tra}(u_i, g_j, t_k) = \arg\min\left(\sum_{i,j}(W_{u_i}\cdot u_i + W_{g_j}\cdot g_j)\times \text{Dist}(u_i, g_j, t_k)\right), \tag{29}$$

where W_u and W_g are weights for UAVs and ground operators, respectively, and $\mathrm{Dist}(u_i, g_j, t_k) = d_{ik} + d_{jk}$ represents the sum of the distances from UAV u_i and ground operator g_j to the task point t_k. This formula accounts for the dynamic adjustment of task allocation, optimizing coordination and resource allocation between UAVs and ground operators to ensure efficient mission completion.

Through the application of these mathematical models, the system achieves balanced resource utilization and maximized task coverage by incorporating real-time data and prioritizing tasks based on their criticality and urgency, as reflected by the criticality factor $U(t_k)$ and the urgency factor $E(t_k)$. The path planning module, leveraging the Voronoi diagram and multi-objective optimization, efficiently minimizes travel distances and navigates obstacles, adapting dynamically to environmental changes. Additionally, the coordination mechanism enhances communication and execution by focusing on critical tasks, ensuring seamless adjustments in response to real-time information. This integrated approach significantly improves task completion rates, resource efficiency and operational safety, ensuring the system's robustness and adaptability in complex and dynamic operational environments.

4. HUCC-MAPPO Algorithm and Implementation

The HUCC-MAPPO algorithm, specifically designed to address the unique challenges of the AGC system, optimizes task allocation and coordination in human–UAV collaborative scenarios. It integrates Voronoi diagram-based path planning, enhancing the precision of task allocation and coordination. The algorithm employs detailed state representations and real-time strategies, significantly enhancing flexibility, coordination efficiency and safety during task execution. By emphasizing the synergy between UAVs and ground operators, it facilitates improved decision-making and comprehensive task coverage, enabling the AGC system to adapt to dynamic environments and execute tasks effectively. Central to this optimization process is a well-defined objective function that quantitatively evaluates key performance metrics, guiding the system toward optimal outcomes.

4.1. *DRL model for HUCC-MAPPO algorithm*

The HUCC-MAPPO algorithm integrates state representation, action sets, reward functions and policy networks to optimize task coordination and path planning within the AGC system. This algorithm is designed to enhance the efficiency of UAVs and ground operators by focusing on minimizing task completion times, maximizing resource utilization, and maintaining operational safety in complex and dynamic environments. The overall structure of the HUCC-MAPPO algorithm, as illustrated in Fig. 5, highlights these integrated components and their roles in achieving the system's objectives.

4.1.1. *State representation*

In the HUCC-MAPPO algorithm, the state representation, denoted as S, is a critical component that captures essential information about the environment, UAVs and ground operators. This representation forms the basis of decision-making in the algorithm by providing a comprehensive snapshot of the system's current status. The state representation is defined as follows:

$$S = \{\boldsymbol{P}_{u_i}, \boldsymbol{v}_{u_i}, \boldsymbol{\Phi}_{u_i}, \boldsymbol{P}_{g_j}, \boldsymbol{\varphi}_{g_j}, d_{ik}, d_{jk}, d_{i,\mathrm{obstacle}}, d_{j,\mathrm{obstacle}}, C_t\}, \tag{30}$$

where $\boldsymbol{P}_{u_i}$, $\boldsymbol{v}_{u_i}$ and $\boldsymbol{\Phi}_{u_i}$ represent the position, velocity and attitude (including pitch, roll and yaw) of UAV u_i. $\boldsymbol{P}_{g_j}$ and $\boldsymbol{\varphi}_{g_j}$ denote the position and current direction angle of ground operator g_j. d_{ik} and d_{jk} represent the distances from UAV u_i and ground operator g_j to the target location t_k, respectively, and are key to minimizing path length in planning. $d_{i,\mathrm{obstacle}}$ and $d_{j,\mathrm{obstacle}}$ represent the distances from UAV u_i and ground operator g_j to the nearest obstacle, ensuring effective obstacle avoidance. C_t represents the current task context information. These state variables collectively inform the decision-making process for task allocation and path planning within the HUCC-MAPPO algorithm, allowing for efficient and safe coordination between UAVs and ground operators.

4.1.2. *Action set*

The action set, denoted as A, in the HUCC-MAPPO algorithm defines the range of possible actions available to UAVs and ground operators during task execution. These actions are fundamental to the algorithm's ability to adapt to the dynamic environment and optimize task performance. The action set is defined as follows:

$$A = \{\Delta\boldsymbol{P}_{u_i}, \Delta\boldsymbol{v}_{u_i}, \Delta\boldsymbol{\Phi}_{u_i}, \Delta\boldsymbol{P}_{g_j}, \Delta\boldsymbol{\varphi}_{g_j}\}, \tag{31}$$

where $\Delta\boldsymbol{P}_{u_i}$ represents the horizontal positional adjustments for UAV u_i, including changes in the x and y coordinates to approach task points t_k or maintain optimal positioning. $\Delta\boldsymbol{v}_{u_i}$ denotes velocity adjustments, allowing modifications to speed in response to task requirements or environmental conditions. $\Delta\boldsymbol{\Phi}_{u_i}$ involves changes in yaw to align the UAV's orientation with the desired direction, while maintaining a constant altitude. For ground operators, $\Delta\boldsymbol{P}_{g_j}$

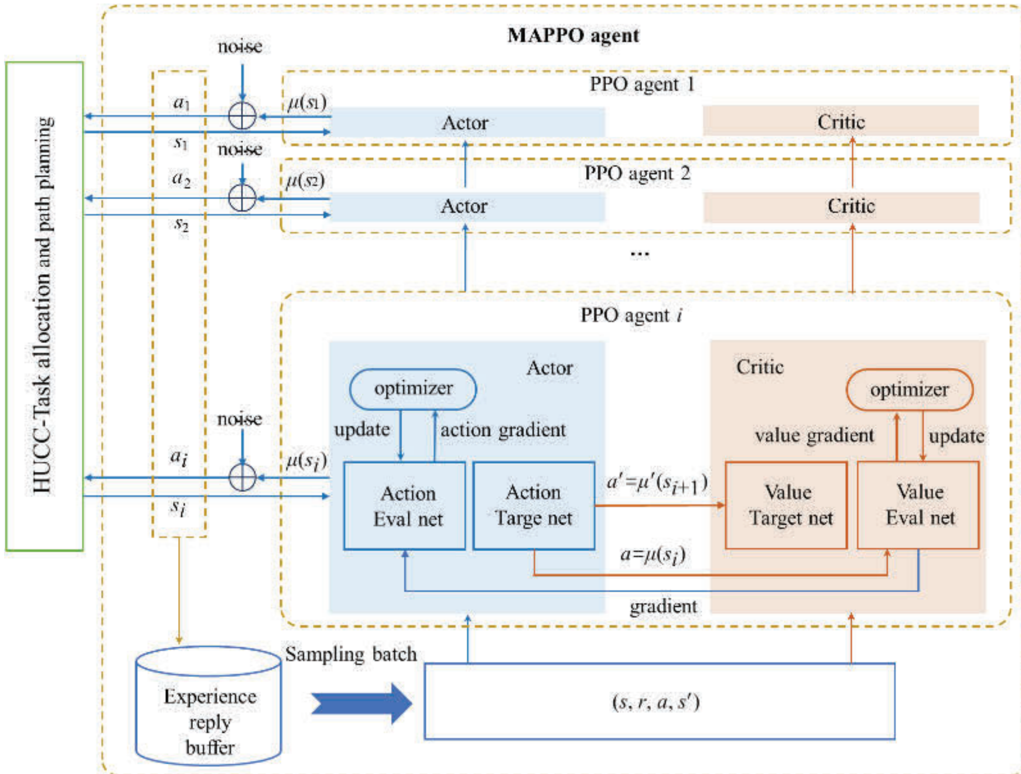


Fig. 5. Schematic diagram of HUCC-MAPPO algorithm structure.

focuses on navigating toward task points t_k while avoiding obstacles, and $\Delta\varphi_{g_j}$ involves directional adjustments to optimize the approach towards task points based on the current orientation.

The directional adjustments for both UAVs and ground operators are calculated to ensure they are oriented toward their respective targets. These calculations are expressed as follows:

$$\Delta\mathbf{\Phi}_{u_i} = \arctan2(y_{t_k} - y_{u_i}, x_{t_k} - x_{u_i})\mathbf{\Phi}_{u_i} \tag{32}$$

$$\Delta\varphi_{g_j} = \arctan2(y_{t_k} - y_{g_j}, x_{t_k} - x_{g_j})\varphi_{g_j} \tag{33}$$

These actions are essential for ensuring that UAVs and ground operators can efficiently navigate towards task points, maintaining optimal positioning and orientation throughout the mission.

4.1.3. *Reward function*

The reward function, denoted as R, is a key component of the HUCC-MAPPO algorithm that evaluates the performance of UAVs and ground operators based on several criteria, including task completion, path length, collision avoidance and coordination efficiency. This function drives the learning process by providing feedback that influences future decisions, ensuring the algorithm continuously improves its effectiveness.

$$R = R_{\text{task}} - R_{\text{path}} - R_{\text{collision}} + R_{\text{coordination}} \tag{34}$$

Task completion reward R_{task} encourages the successful completion of tasks:

$$R_{\text{task}} = \sum_{k=1}^{K} r_{\text{complete}}(t_k) \tag{35}$$

where r_{complete} is the reward value for completing a task t_k. The reward is set according to the task's difficulty, importance, and urgency. For instance, simple tasks may have $r_{\text{complete}}(t_k) = 5$, moderate tasks $r_{\text{complete}}(t_k) = 10$ and difficult tasks $r_{\text{complete}}(t_k) = 15$.

The values for r_{complete} are determined based on predefined metrics that quantify task difficulty, importance and urgency. These metrics play a crucial role in task allocation and execution, ensuring that the system adapts to real-world scenarios.

The path length penalty R_{path} is designed to penalize longer paths, thus encouraging shorter and more efficient routes:

$$R_{\text{path}} = \sum_{i=1}^{M}\sum_{k=1}^{K} a_{ik} \cdot d_{ik} + \sum_{j=1}^{N}\sum_{k=1}^{K} g_{jk} \cdot d_{jk} \tag{36}$$

This penalty aims to minimize the total distance traveled by UAVs and ground operators, reducing both time and energy consumption. By controlling path lengths, the system can more effectively allocate resources and decrease uncertainties in task execution.

The collision avoidance penalty $R_{\text{collision}}$ is applied to prevent path conflicts between UAVs and ground operators:

$$R_{\text{collision}} = \sum_{i \neq m} \text{collision}(u_i, u_m) + \sum_{j \neq m} \text{collision}(g_j, g_n) \quad (37)$$

where the collision function for UAVs u_i and u_m, or ground operators g_j and g_n are defined based on distance and direction angle thresholds:

$$\text{collision}(u_i, u_m) = \begin{cases} 20 & \text{if } d_{im} \leq d_{\text{threshold\_}u} \textit{ and } \Phi_{im} \leq \Phi_{\text{threshold}}, \\ 0, & \text{otherwise.} \end{cases} \quad (38)$$

$$\text{collision}(g_j, g_n) = \begin{cases} 40 & \text{if } d_{jn} \leq d_{\text{threshold\_}g} \textit{ and } \varphi_{jn} \leq \varphi_{\text{threshold},} \\ 0 & \text{otherwise.} \end{cases} \quad (39)$$

The thresholds $d_{\text{threshold\_}u} = 15$ and $d_{\text{threshold\_}g} = 1.5$, as well as angle thresholds $\varphi_{\text{threshold}} = 10$, and $\Phi_{\text{threshold}} = 10$, are set to ensure safe and practical operating conditions.

These thresholds are based on simulation and empirical data, ensuring safe separation between UAVs and ground operators in various operational scenarios. Adjustments to these thresholds may be necessary depending on the specific task environment.

The coordination efficiency reward $R_{\text{coordination}}$ encourages effective coordination between UAVs and ground operators:

$$R_{\text{coordination}} = \text{coordination}(u_i, g_j) \times \text{reward}_{\text{coordination}} \times \text{eff}_{\text{total}}, \quad (40)$$

where the efficiency for each task is calculated as follows:

$$\text{eff}_k = \frac{1}{d_{ik} + d_{jk} + \epsilon}, \quad (41)$$

where ϵ is a small positive constant to prevent division by zero.

Calculating the overall coordination efficiency involves summing the efficiencies across all tasks:

$$\text{eff}_{\text{total}} = \frac{1}{K} \sum_{k=1}^{K} \text{eff}_k. \quad (42)$$

Determining if the coordination efficiency meets the threshold is done as follows:

$$\text{coordination}(u_i, g_j) = \begin{cases} 1 & \text{if eff}_{\text{total}} \geq \text{eff}_{\text{threshold},} \\ 0 & \text{otherwise,} \end{cases} \quad (43)$$

where reward $R_{\text{coordination}} = 10$ is a weight coefficient for the coordination efficiency reward.

The coordination efficiency is closely tied to the distances and weights defined earlier. By minimizing these distances and maximizing coordination efficiency, the system fosters optimal collaboration between UAVs and ground operators.

In summary, the coordination efficiency reward $R_{\text{coordination}}$ is influenced by the distances between UAVs, ground operators and targets, as well as the safe distance constraints in Eq. (26). This constraint requires ground operators to maintain a safe distance from obstacles, which affects their path choices and thus impacts coordination efficiency. While maintaining safe distances ensures smooth and collision-free movement, it may also increase path lengths, affecting the efficiency score. Overall, this approach supports effective collaboration between UAVs and ground operators, aligning with the goals of the HUCC-MAPPO algorithm.

4.1.4. *Policy network*

In the HUCC-MAPPO algorithm, the policy network is designed to generate and implement optimized action strategies that enable UAVs and ground operators to efficiently and effectively coordinate tasks in dynamic and complex environments. To enhance the performance, a hierarchical decision-making framework with shared policy networks is adopted. This framework integrates task allocation, path planning and coordination control, thereby reducing the search space for policies and improving the robustness of action strategy generation.

The policy network employs a deep neural network (DNN) [39, 40], where the input layer processes the state representation S, and the output layer generates the action probability distribution $\pi_\theta(a|s)$ through a series of linear transformations and nonlinear activations. ReLU activation functions are used in the hidden layers to capture complex state-action relationships, prevent vanishing gradient issues, and ensure effective decision-making.

The action probability distribution $\pi_\theta(a|s)$ is generated as follows:

$$\pi_\theta(a|s) = \text{softmax}(\boldsymbol{W}_2 \cdot \text{ReLU}(\boldsymbol{W}_1 \cdot S + b_1) + b_2), \quad (44)$$

where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are weight matrices, and b_1 and b_2 are bias terms. $\boldsymbol{W}_1$ and b_1 map the state set S to an intermediate hidden representation, while $\boldsymbol{W}_2$ and b_2 map this hidden representation to the action space. The weights and biases are trained using the proximal policy optimization (PPO) algorithm, allowing the network to adapt effectively to different task environments.

Training of the policy network is conducted using the PPO algorithm, which stabilizes the learning process by constraining large policy updates. The optimization involves gradient-based backpropagation of the objective function, with hyperparameters such as learning rate and batch size adjusted to improve efficiency and convergence. In a multi-agent environment, the policy network coordinates actions through parameter sharing and joint optimization, enabling effective cooperation between UAVs and ground operators. This coordination ensures adaptability to dynamic environments. The network's performance is evaluated based on metrics such as task completion rate, path planning efficiency and coordination efficiency, thereby maintaining stability and effectiveness across various task scenarios.

4.1.5. *Policy gradient*

The HUCC-MAPPO algorithm optimizes the policy network using a policy gradient method derived from MAPPO, which is an extension of PPO designed for multi-agent systems. PPO is particularly advantageous in dynamic environments like those involving UAVs and ground operators due to its ability to balance exploration and exploitation while maintaining stability through constrained policy updates. This ensures that the HUCC-MAPPO algorithm can adapt to real-time changes effectively, optimizing coordination strategies and task performance.

The policy gradient in the HUCC-MAPPO algorithm is calculated using the following equation:

$$\nabla_\theta J(\theta) = \mathbb{E}_t[\nabla_\theta \log \pi_\theta(a_t|s_t)(\sum_{t'=t}^{T} \gamma^{t'-t} r_{t'} - V_\phi(s_t))] \quad (45)$$

where the policy function $\pi_\theta(a_t|s_t)$ represents the probability of selecting action a_t given state s_t, which guides the action selection process. $\pi_\theta(a_t|s_t)$ represents the probability of selecting action a_t given state s_t, which guides the action selection process. $r_{t'}$ is the reward received at time step t', accumulated over time to evaluate the policy's long-term effectiveness. The discount factor $\gamma(0 < \gamma \leq 1)$ adjusts future rewards to reflect their present value, balancing short-term and long-term rewards. $\sum_{t'=t}^{T} \gamma_{t'}^{t'-tr}$ represents the cumulative discounted reward from time step t to T, reflecting the total expected rewards over the entire horizon. $V_\phi(s_t)$ is the value function that estimates the expected cumulative reward starting from state s_t, acting as a baseline to reduce variance and stabilize training.

The policy gradient equation is crucial in updating the policy network's parameters θ. It directs the learning process by adjusting θ to increase the probability of actions that lead to higher cumulative rewards. The equation balances the exploration of new actions with the exploitation of known rewarding actions, ensuring that the policy gradually improves over time. This gradient is computed during training and is used to optimize the policy network by minimizing the difference between the expected reward (estimated by the value function) and the actual cumulative reward. The process iteratively refines the policy to achieve optimal task performance in the dynamic environment where UAVs and ground operators operate.

The value function is defined as follows:

$$V_\phi(s_t) = \mathbb{E}\left[\sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}|s_t\right]. \quad (46)$$

This function estimates the expected return from state s_t, helping to stabilize the training process by providing a consistent baseline.

In addition to optimizing the policy network, the value network V_ϕ is trained to minimize the value loss function, which is defined as follows:

$$L^{\text{value}}(\theta) = \mathbb{E}_t[(r_t - V_\phi(s_t))^2]. \quad (47)$$

This value loss function helps to refine the value network's predictions by minimizing the discrepancy between the estimated value V_ϕ and the actual cumulative rewards R_t. Optimizing this loss function ensures that the value network provides a more accurate baseline for the policy gradient, thereby stabilizing and improving the training process.

PPO optimizes the policy network by constraining updates, ensuring that new policies do not deviate excessively from the old ones, which is crucial in dynamic environments. This clipping mechanism is defined as follows:

$$\nabla_\theta L^{\text{CLIP}}(\theta) = \mathbb{E}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)], \quad (48)$$

where $r_t(\theta)$ is the probability ratio between the updated and old policies, and $\hat{A}_t$ is the advantage estimate, stabilizes the training by preventing large, potentially destabilizing policy changes. This method allows the HUCC-MAPPO algorithm to maintain a consistent and reliable performance in environments that require frequent adaptations, such as those involving UAV and ground operator coordination.

The training process also involves optimizing hyperparameters, such as learning rate, clip range and batch size, to enhance convergence and performance. These optimizations are crucial for ensuring that the policy network adapts effectively to the dynamic and complex environments in which UAVs and ground operators must collaborate.

Overall, the use of PPO in the HUCC-MAPPO algorithm facilitates robust and stable learning, enabling effective coordination and collaboration between UAVs and ground operators in dynamic and complex environments. The policy gradient method ensures that the policy network continually improves, optimizing the system's overall performance in various task scenarios.

4.2. ***HUCC-MAPPO algorithm implementation and operational logic***

The HUCC-MAPPO algorithm is structured to optimize both task allocation and path planning within the AGC system by leveraging advanced coordination strategies and continuous learning mechanisms. This section details the specific implementation steps of the algorithm, including the collection of environmental observations, dynamic task prioritization, action execution and the iterative updating of policy and value functions. The algorithm's implementation emphasizes the importance of real-time coordination between UAVs and ground operators, adapting to the evolving operational environment to maximize system performance.

4.2.1. *Learning rate and discount factor*

The initial learning rate is set to 0.001 to balance convergence speed and model stability. Higher learning rates may accelerate learning but can cause instability. The selected value of 0.001 ensures both efficient convergence and stable performance across different environments. The discount factor γ is set to 0.99, ensuring that the system considers future rewards to enhance long-term objectives, which showed superior performance compared to lower discount factors in experiments.

4.2.2. *Exploration rate and decay*

The exploration rate is initially set to 1.0, with a decay rate of 0.995 and a minimum exploration rate of 0.01. This ensures sufficient exploration during the initial learning stages, while the gradual decay allows the UAVs to transition toward exploiting optimal strategies as learning progresses. A balanced decay rate of 0.995 was found to effectively stabilize the system and speed up convergence.

4.2.3. *Neural network structure*

The neural network structure used by the HUCC-MAPPO algorithm consists of a two-layer fully connected network with 128 nodes per layer. This configuration balances computational efficiency and learning capability, providing the ability to generalize effectively across different task scenarios. The structure enables efficient decision-making and task execution, particularly in multi-task, dynamic environments.

4.2.4. *Parameter selection rationale*

The rationale behind parameter choices is detailed to maintain a balance between exploration and exploitation while ensuring stability during training. For instance, the learning rate, discount factor and exploration rate were selected based on iterative experiments to ensure an effective trade-off between speed and stability during convergence. These selections also provide resilience in varying mission environments.

4.2.5. *HUCC-MAPPO algorithm pseudo-code*

The following pseudo-code provides a comprehensive overview of the key steps involved in the implementation of the HUCC-MAPPO algorithm, encompassing the initial parameter setup, execution of coordinated actions and iterative policy updates. This pseudo-code serves as a practical framework for deploying the HUCC-MAPPO algorithm within dynamic and complex environments, emphasizing the critical processes essential for achieving effective coordination between UAVs and ground operators.

Algorithm 1. HUCC-MAPPO algorithm pseudo-code.

Initialize policy network parameters θ and value network parameters ϕ
Initialize replay buffer for storing experience tuples
for episode = 1 to M do
  Reset environment and obtain initial state s_0.
  for t = 1 to T do
    for each agent i (UAVs) do
      Observe current state s_t and calculate distances d_{ik}, d_{jk}.
      Complexity: $O((M+N)^2)$ for calculating distances between agents
      Determine action a_t using policy $\pi_\theta(a_t|s_t)$.
      Execute action a_t in the environment.
      Receive reward r_t and observe next state ${s'}_t$.
      Store transition $(s_t, a_t, r_t, {s'}_t)$ in replay buffer.
    end for
    if replay buffer is full then
      for each mini-batch do
        Sample mini-batch of transitions from replay buffer.
        Compute advantage estimates $\hat{A}_t$ using value network $V_\phi(s_t)$.
        Update policy network θ using PPO objective:
        $\nabla_\theta J(\theta) = \mathbb{E}_t[\nabla_\theta \log \pi_\theta(a_t|s_t) \left(\sum_{t'=t}^{T} \gamma^{t'-t} R_{t'} - V_\phi(s_t)\right)\Big]$.
        Update value network ϕ by minimizing value loss:
        $L^{\text{value}}(\theta) = \mathbb{E}_t[(R_t - V_\phi(s_t))^2]$.
      end for
      Clear replay buffer.
    end if
  end for
end for

5. Experimental Design and Evaluation

5.1. *Experimental environment*

To validate the effectiveness and robustness of the proposed HUCC-MAPPO algorithm, experiments were conducted in a simulated environment on a computer equipped with an Intel i9 processor, 32 GB of RAM and an NVIDIA RTX 3080 GPU. This high-performance hardware setup ensured smooth operation of the simulations and the ability to handle the complex computations required by the algorithm. The software implementation was done using Python as the primary programming language, with TensorFlow and PyTorch frameworks utilized for their computational efficiency and flexibility in machine learning tasks. The simulation environment was developed using the open-source UAV simulation platform AirSim. The experimental field was meticulously designed to replicate various real-world scenarios, as illustrated in Fig. 6.

Multiple task scenarios were crafted to evaluate the performance of the HUCC-MAPPO algorithm across various UAV operational dimensions. The task area was divided into multiple subregions using Voronoi diagrams, ensuring organized partitioning based on the spatial distribution of UAVs and ground operators. Each subregion represented distinct task types and environmental conditions, testing different aspects of the algorithm's capabilities. Simple regions, such as open plains and basic obstacle areas, were used to measure the algorithm's baseline task completion rates and path planning capabilities. Static obstacles, such as buildings and barriers, were added to the experimental field to test the UAVs and ground operators' obstacle avoidance capabilities. These obstacles provided a controlled environment for evaluating the system's effectiveness in recognizing and avoiding fixed obstacles, thus enhancing the realism and complexity of the experimental scenarios.

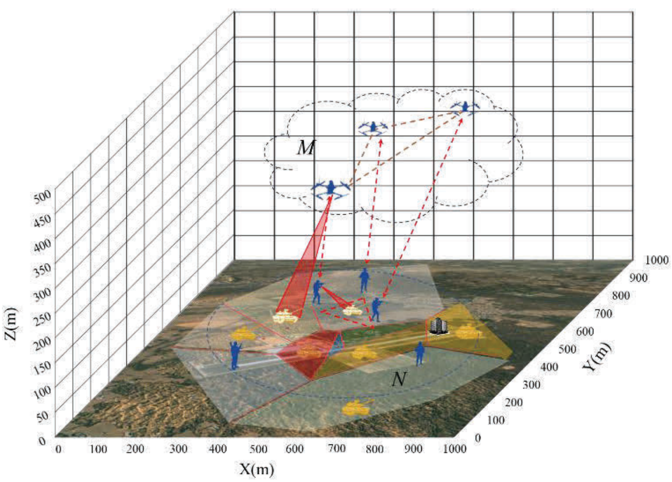


Fig. 6. Schematic diagram of the experimental field.

5.2. *Parameter settings*

To ensure the rigor and reliability of the experiments, key parameters were meticulously set and categorized into task scenario parameters and HUCC-MAPPO algorithm parameters, as detailed in Tables 1 and 2.

A typical operational scenario involved a task force consisting of 25 ground operators and 50 UAVs, executing reconnaissance missions across 50 task points within a $1000\,\text{m} \times 1000\,\text{m} \times 500\,\text{m}$ area. Each operator and UAV was assigned up to two task points, taking into account the task load capacity. To optimize communication resources and enhance control efficiency, the communication bandwidth between UAVs and ground operators was limited to 4 Mbps. Each UAV was equipped with visible light reconnaissance payloads and a combined GPS and IMU navigation system to enhance navigation accuracy.

The HUCC-MAPPO algorithm parameters were configured to optimize the training and performance of the neural network. These parameters were designed to fine-tune the learning process, ensuring that the algorithm could

Table 1. Task scenario parameters.

Parameter	Value
Number of UAVs	50
Number of ground operators	25
Number of tasks	50
Number of obstacles	50 (including 40 static obstacles, 5 radar zones, 5 artillery threat zones)
Test aera range	$1000\,\text{m} \times 1000\,\text{m} \times 500\,\text{m}$
Maximum task range	2 per UAV and ground operator
Maximum bandwidth	4 Mbps
Maximum transmission distance	100 m

Table 2. HUCC-MAPPO algorithm parameters.

Parameter	Value
Learning rate	0.001
Discount factor	0.99
Training iterations	5000
Neural network structure	Two-layer fully connected network with 128 nodes per layer
Batch size	64
Optimizer	Adam
Initial exploration rate	1.0
Exploration rate decay	0.995
Minimum exploration rate	0.01
Clipping range	0.2
Value function coefficient	0.5
Entropy coefficient	0.01

effectively adapt to various task scenarios and environmental changes. By systematically adjusting these parameters, the experiments aimed to assess the algorithm's learning efficiency and its ability to generalize across different situations.

5.3. *Evaluation metrics*

To evaluate the performance and robustness of the HUCC-MAPPO algorithm, a set of key evaluation metrics was employed. These metrics quantify the algorithm's efficiency, coordination and adaptability, providing insights into its behavior in various complex environments and informing future improvements. Specifically, for obstacle avoidance evaluation, multiple metrics were employed to capture the system's performance in avoiding static obstacles. This includes tracking successful obstacle avoidance events, measuring minimum safe distances to obstacles and evaluating the effectiveness of avoidance paths.

(1) Task coverage efficiency

Task coverage efficiency (TCE) builds upon the resource utilization metrics defined earlier in Eqs. (17) and (18). While those equations measure the resource utilization rates $R(u_i)$ and $R(g_j)$ for UAVs and ground operators, respectively, TCE expands this concept to quantify the overall coverage efficiency across the mission. It reflects how well the system avoids redundant coverage and omissions, and can be mathematically expressed as follows:

$$\mathrm{TCE} = \frac{\sum_{i=1}^{M}\sum_{k=1}^{K} D_{ik}\cdot\frac{r_{ik}}{\max(r_{ik})} + \sum_{j=1}^{N}\sum_{k=1}^{K} D_{jk}\cdot\frac{r_{jk}}{\max(r_{jk})}}{N_{\text{total}}} \times 100\%, \tag{49}$$

where N_{total} represents the total number of mission points designated in the mission planning.

(2) Computation time

Computation time (CT) refers to the total time required for computational operations, such as decision-making, path planning, and obstacle avoidance, and is used to evaluate the algorithm's real-time performance. On the other hand, task completion time (TCT) represents the time required for UAVs and ground operators to execute tasks, including moving to target locations and completing assigned tasks. These two metrics, considered together, provide a comprehensive evaluation of the system's computational efficiency and overall mission execution efficiency.

$$\mathrm{CT} = \sum_{i=1}^{M}\sum_{k=1}^{K} D_{ik}\cdot\tau_{ik} + \sum_{j=1}^{N}\sum_{k=1}^{K} D_{jk}\cdot\tau_{jk}. \tag{50}$$

(3) Path length

Path length (PL) measures the total distance traveled by UAVs and ground operators to complete the assigned tasks. The path planning model is designed to minimize the total path length to enhance task execution efficiency. The path lengths d_{ik} and d_{jk} for UAV u_i and ground operator g_j to the task points t_k have been defined in Eqs. (2), (3) and (20), respectively. The total path length can be mathematically expressed as follows:

$$\mathrm{PL} = \sum_{i=1}^{M}\sum_{k=1}^{K} d_{ik} + \sum_{j=1}^{N}\sum_{k=1}^{K} d_{jk}, \tag{51}$$

where d_{ik} is the distance covered by UAV u_i for task t_k, and d_{jk} is the distance covered by ground operator g_j for task t_k.

(4) Obstacle avoidance capability

Obstacle avoidance capability (OAC) is a critical metric that assesses the ability of UAVs and ground operators to navigate through the mission area while successfully avoiding obstacles. OAC can be mathematically expressed as follows:

$$\mathrm{OAC} = 1 - \frac{N_{\text{collision}}}{N_{\text{encounter}}}, \tag{52}$$

where $N_{\text{collision}}$ represents the number of collisions with obstacles that occurred during the mission, and $N_{\text{encounter}}$ represents the total number of obstacle encounters recorded during the mission. A lower value of $N_{\text{collision}}$ relative to $N_{\text{encounter}}$ indicates better obstacle avoidance performance.

(5) Task completion rate

Task completion rate (TCR) is calculated as the ratio of successfully completed tasks to the total number of tasks. This metric provides insight into the overall performance and efficiency of the AGC system in accomplishing mission objectives. The TCR can be mathematically calculated using the following formula:

$$\mathrm{TCR} = \frac{\sum_{k=1}^{K} \delta(t_k)}{K}, \tag{53}$$

where $\delta(t_k) = 1$ if task t_k is completed, otherwise $\delta(t_k) = 0$, and K is the total number of tasks.

To further refine the calculation of TCR, the concept of task weighting based on priority and urgency is introduced. The weighted TCR (WTCR) can be mathematically expressed as follows:

$$\mathrm{WTCR} = \frac{\sum_{k=1}^{K} w_k \delta(t_k)}{\sum_{k=1}^{K} w_k}, \tag{54}$$

where w_k represents the weight assigned to each task t_k based on its priority and urgency, reflecting the importance of the task in the overall mission objectives. This weighted measure provides a more nuanced view of system performance, emphasizing the successful completion of high-priority tasks.

These evaluation metrics enable a comprehensive assessment of the HUCC-MAPPO algorithm's performance, offering a deep understanding of its capabilities and highlighting areas for future improvement.

5.4. *Experimental results*

To validate the effectiveness and robustness of the proposed HUCC-MAPPO algorithm, comparative experiments were conducted against the heuristic greedy (HG) algorithm, the mixing Q-network (QMIX) algorithm, the multi-agent deep deterministic policy gradient (MADDPG) algorithm and the MAPPO algorithm. The HG algorithm is widely utilized for task allocation and path planning due to its low-computational complexity and straightforward implementation [41]. It incrementally builds solutions by selecting the most advantageous options, making it suitable for environments with limited computational resources. In contrast, the QMIX algorithm enhances multi-agent RL by optimizing agent coordination through a mixing network that combines individual Q-values into a global Q-value [42], making it effective for dynamic adjustments and efficient resource utilization. The MAPPO algorithm employs policy networks and the PPO method to ensure stable and robust policy updates [43], facilitating coordinated and adaptive behaviors among agents in complex environments. The MADDPG algorithm, on the other hand, adopts a centralized training and decentralized execution framework. This allows each agent to learn its own policy while benefiting from centralized critic information, making it well-suited for complex, continuous action space environments. MADDPG excels in optimizing coordination and decision-making among multiple UAVs but may face scalability issues when applied to large-scale scenarios [44]. This makes MAPPO an ideal benchmark for evaluating the improvements of the HUCC-MAPPO algorithm in human–machine coordination and path optimization.

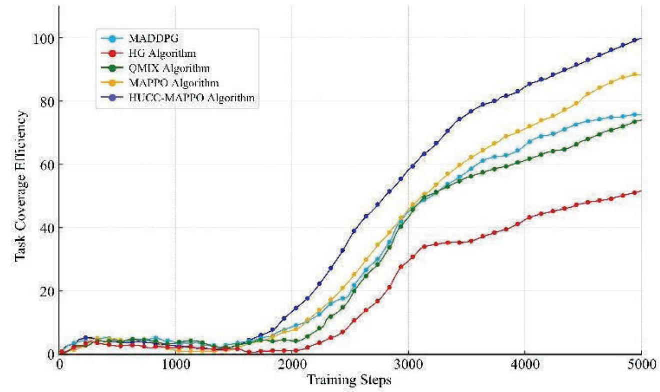


Fig. 7. TCE comparison of different algorithms.

(1) TCE analysis

The HUCC-MAPPO algorithm significantly enhanced TCE in this experimental scenario, increasing it by over 40%, see Fig. 7. The HG algorithm showed the lowest efficiency due to its lack of advanced coordination. The QMIX algorithm improved TCE but did not reach the level achieved by the HUCC-MAPPO algorithm. The MAPPO algorithm also performed well, demonstrating significant improvements over both the HG and QMIX algorithms. The MADDPG algorithm demonstrated moderate improvement, performing slightly better than QMIX but below MAPPO and HUCC-MAPPO.

(2) CT analysis

The HG algorithm, owing to its simplicity, achieved the shortest CTs but performed poorly in terms of optimization and overall efficiency, see Fig. 8. The QMIX and MAPPO algorithms exhibited increased CTs owing to their more complex optimization mechanisms, yet they achieved better performance outcomes than the HG algorithm. However, the HUCC-MAPPO algorithm excelled in balancing computational efficiency with performance improvements, showing higher CTs but maintaining superior optimization capabilities. The MADDPG algorithm exhibited moderate CTs,

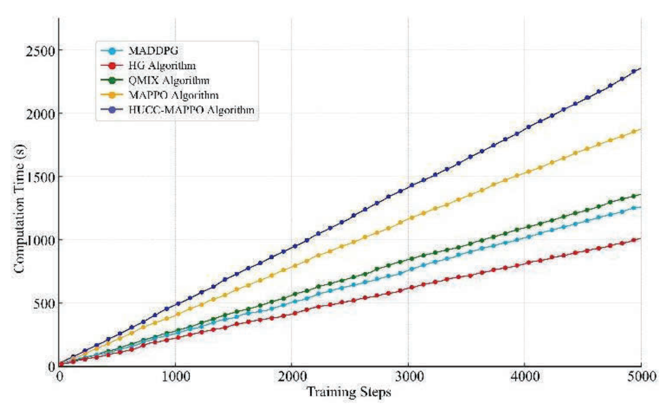


Fig. 8. CT comparison of different algorithms.

higher than HG but lower than QMIX, with efficiency still falling short of HUCC-MAPPO.

(3) PL analysis

The HUCC-MAPPO algorithm significantly reduced the PLs of UAVs and ground operators, achieving over a 50% reduction compared to traditional algorithms. The HG algorithm, with its simplistic approach, resulted in the longest paths. QMIX demonstrated some improvement but fell short of HUCC-MAPPO's efficiency. MAPPO achieved notable reductions but was still slightly less effective than HUCC-MAPPO. The MADDPG algorithm reduced path lengths compared to QMIX but was slightly less effective than MAPPO and HUCC-MAPPO, see Fig. 9.

(4) OAC analysis

In environments with obstacles, the HUCC-MAPPO algorithm effectively avoided collisions, achieving a success rate of over 95%. The HG algorithm performed poorly in these environments, with a success rate of only around 45%. The QMIX and MAPPO algorithms demonstrated notable improvements in obstacle avoidance, achieving success rates of approximately 65% and 75%, respectively. The MADDPG algorithm achieved a success rate of around 70%, slightly below MAPPO, see Fig. 10.

(5) TCR analysis

The HUCC-MAPPO algorithm consistently achieved a high TCR in this experimental scenario, often approaching 100% in complex environments, indicating near-optimal performance. In contrast, the HG algorithm exhibited a lower completion rate, particularly under complex conditions, achieving approximately 39% on average. The QMIX and MAPPO algorithms performed better in complex environments, achieving completion rates of approximately 58% and 80%, respectively. The MADDPG algorithm achieved a TCR of about 60%, between QMIX and MAPPO, see Fig. 11.

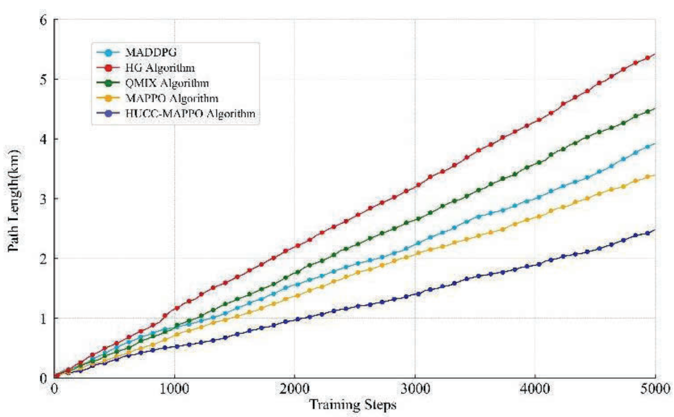


Fig. 9. PL comparison of different algorithms.

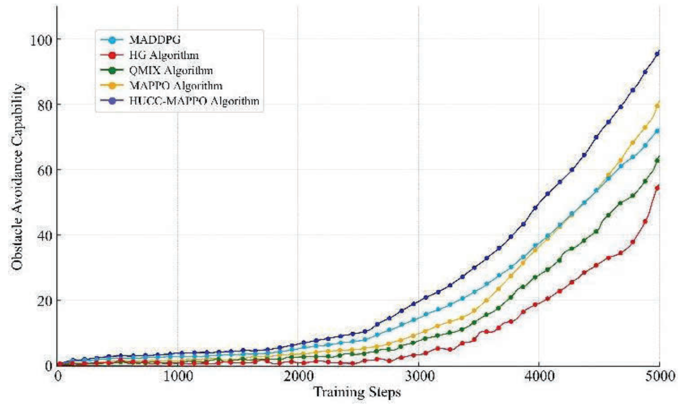


Fig. 10. OAC comparison of different algorithms.

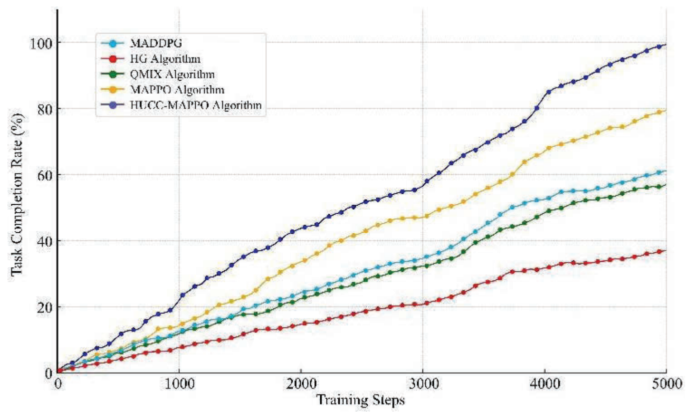


Fig. 11. TCR comparison of different algorithms.

(6) Learning curve analysis

To evaluate the performance of the four algorithms within the AGC system, particularly regarding their convergence and stability over extended training periods, we conducted a learning curve analysis. These curves illustrate the dynamic changes in each algorithm's performance before reaching a stable state, thereby facilitating the identification of the most suitable algorithm for practical applications, see Fig. 12.

The HUCC-MAPPO algorithm demonstrated the fastest convergence and the highest stability, making it particularly well-suited for the AGC system. The MAPPO algorithm showed good convergence and stability, with final performance slightly below that of HUCC-MAPPO. The MADDPG algorithm demonstrated reasonably fast convergence with good stability, consistently outperforming QMIX but still lagging behind the HUCC-MAPPO and MAPPO algorithms in terms of final performance. It showed some promising initial results, but its learning rate slowed down in later stages, indicating limited capability in handling the complex dynamics compared to HUCC-MAPPO and MAPPO. The

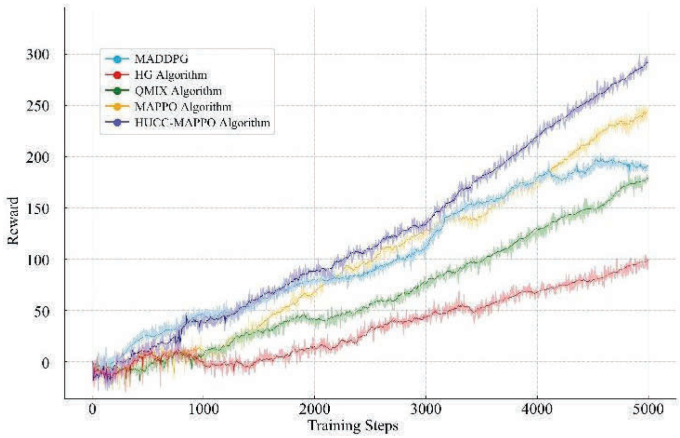


Fig. 12. Learning curve comparison of different algorithms.

QMIX algorithm exhibited moderate performance, with significant fluctuations during training, suggesting some instability. The HG algorithm performed the worst, with considerable performance fluctuations and inadequate outcomes, indicating poor convergence and limited suitability for practical applications. Overall, HUCC-MAPPO was the most effective, followed by MAPPO, MADDPG, QMIX and HG.

Comparative experiments indicate that the HUCC-MAPPO algorithm offers significant advantages in task completion rate, path length, coordination efficiency, obstacle avoidance capability and computation time. In particular, the HUCC-MAPPO algorithm performed effectively in complex tasks and dynamic environments, reducing both path length and computation time more efficiently. The HG algorithm, while having lower computational complexity, performed the least effectively in most scenarios, especially in complex environments. The QMIX algorithm showed moderate improvements across various metrics but faced challenges in stability and efficiency under dynamic conditions. The MAPPO algorithm achieved strong results in coordination and path length but was slightly less efficient compared to HUCC-MAPPO in complex scenarios. The MADDPG algorithm demonstrated comparable results to MAPPO in several metrics, performing well in coordination efficiency and obstacle avoidance, but was not as effective as HUCC-MAPPO in adapting to highly dynamic environments. To further illustrate the performance of different algorithms, a comparison of key metrics across different algorithms is presented in Table 3.

Table 3. Comparison of the performance of different algorithms.

Algorithm	TCE	CT	PL	OAC	TCR
HG	Low	Low	Long	45%	39%
QMIX	Medium	Medium	Medium	65%	58%
MADDPG	Medium	Medium	Medium	70%	60%
MAPPO	High	High	Short	75%	80%
HUCC-MAPPO	High	High	Short	95%	100%

Notes: TCE: Task coordination efficiency, CT: Computation time, PL: Path length, OAC: Obstacle avoidance capability, TCR: Task completion rate.

6. Conclusion and Future Work

This study proposed and implemented an AGC system based on the HUCC-MAPPO algorithm to enhance task area coverage and execution effectiveness between UAVs and ground operators. The algorithm extends the existing MAPPO framework by incorporating Voronoi diagram-based path planning to improve the assessment of human–UAV coordination efficiency. Simulation results indicate that the HUCC-MAPPO algorithm significantly improves the efficiency of task area coverage, reduces the time required for computations and the overall length of the planned paths, enhances the system's ability to avoid obstacles, increases the rate of successful task completions and accelerates the learning process when compared to traditional methods.

The HUCC-MAPPO algorithm integrates adaptive task allocation, precise path planning and multi-source information processing, strengthening resource management and decision-making in complex environments. This comprehensive approach enhances the AGC system's ability to handle diverse mission requirements and improve overall operational effectiveness.

Despite these achievements, several limitations remain. First, the experiments were primarily conducted in simulated environments and are yet to be validated in real-world settings, raising questions about the algorithm's practical applicability and robustness under diverse conditions. Second, the HUCC-MAPPO algorithm's computational complexity is relatively high, which may hinder real-time performance, particularly in large-scale tasks and multi-agent systems. Additionally, the algorithm's performance heavily depends on parameter settings, requiring careful tuning for different scenarios, which may limit its generalizability and ease of deployment.

Future research should address these limitations by validating the proposed algorithm in real-world environments to assess its performance and applicability in actual tasks, including scenarios that simulate the complexities and uncertainties of operational conditions. Further optimization is essential to reduce the computational complexity of the HUCC-MAPPO algorithm, enhancing its suitability for real-time applications. Additionally, expanding comparative experiments by incorporating more advanced coordination algorithms and conducting sensitivity analyses of key parameters will help improve system adaptability and performance in multi-agent scenarios. Moreover, the inclusion of performance numbers from our experimental results indicates that the HUCC-MAPPO algorithm achieved a task completion rate improvement of approximately 20%, a reduction in path length by 15% and a 95% success rate in obstacle avoidance compared to other algorithms. Furthermore, enhancing the simulation of real-world obstacles, including dynamic threats and more complex terrain features, will provide a more comprehensive evaluation of the algorithm's effectiveness in rapidly changing environments. These improvements and optimizations are expected to make the AGC system more efficient, reliable and applicable in future applications, reinforcing its potential as a robust solution for advanced human–UAV coordination in complex environments.

ORCID

Zhengyang Cao https://orcid.org/0000-0002-3869-5771
Gang Chen https://orcid.org/0000-0002-5145-5529

References

[1] Z. T. Wei and X. L. Zhao, Multi-UAVs cooperative reconnaissance task allocation under heterogeneous target values, *IEEE Access* **10** (2022) 70955–70963.

[2] F. Yan, J. Chu, J. W. Hu and X. P. Zhu, Cooperative task allocation with simultaneous arrival and resource constraint for multi-UAV using a genetic algorithm, *Expert Syst. Appl.* **245** (2024) 123023.

[3] P. Yao and X. Wei, Multi-UAV information fusion and cooperative trajectory optimization in target search, *IEEE Syst. J.* **16**(3) (2022) 4325–4333.

[4] M. Mozaffari, W. Saad, M. Bennis and M. Debbah, Mobile unmanned aerial vehicles (UAVs) for energy-efficient internet of things communications, *IEEE Trans. Wirel. Commun.* **16**(11) (2017) 7574–7589.

[5] Y. J. Chen, D. K. Chang and C. Zhang, Autonomous tracking using a swarm of UAVs: A constrained multi-agent reinforcement learning approach, *IEEE Veh. Technol. Mag.* **69**(11) (2020) 13702–13717.

[6] N. Zhao, W. D. Lu, M. Sheng, Y. F. Chen, J. Tang, F. R. Yu and K. K. Wong, UAV-assisted emergency networks in disasters, *IEEE Wirel. Commun.* **26**(1) (2019) 45–51.

[7] K. Mase and H. Okada, Message communication system using unmanned aerial vehicles under large-scale disaster environments, in *26th IEEE Int. Symp. Pers. Indoor Mob. Radio Commun.* (IEEE, 2015), pp. 2171–2176.

[8] S. Y. Xu, J. Zhang, S. M. Meng and J. Xu, Task allocation for unmanned aerial vehicles in mobile crowdsensing, *Wirel. Netw.* **30** (2024) 3707–3719.

[9] H. B. Ouyang, L. S. Liang, S. Li and L. Q. Gao, A joint optimization method for multi-UAV deployment and task scheduling in mobile edge computing with large-scale mobile users, *Expert Syst. Appl.* **255** (2024) 124526.

[10] L. Sun, J. S. Wang, L. T. Wan, K. X. Li, X. J. Wang and Y. Lin, Human-UAV interaction assisted heterogeneous UAV swarm scheduling for target searching in communication denial environment, *IEEE Trans. Autom. Sci. Eng.* (2024) 1–16.

[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, Massachusetts, 1998).

[12] Z. Ghahramani, Probabilistic machine learning and artificial intelligence, *Nature* **521** (2015) 452–459.

[13] S. Gronauer and K. Diepold, Multi-agent deep reinforcement learning: A survey, *Artif. Intell. Rev.* **55** (2022) 895–943.

[14] Y. Matsuo, Y. LeCun, M. Sahani, D. Precup, D. Silver, M. Sugiyama, E. Uchibe and J. Morimoto, Deep learning, reinforcement learning, and world models, *Neural Netw.* **152** (2022) 267–275.

[15] C. Dai, K. Zhu and E. Hossain, Multi-agent deep reinforcement learning for joint decoupled user association and trajectory design in full-duplex multi-UAV networks, *IEEE Trans. Mob. Comput.* **22**(10) (2023) 6056–6070.

[16] C. Y. Zhang, Z. K. Li, C. L. He, K. Z. Wang and C. H. Pan, Deep reinforcement learning based trajectory design and resource allocation for UAV-assisted communications, *IEEE Commun. Lett.* **27**(9) (2023) 2398–2402.

[17] J. Liu and Y. P. Zhao, Role-oriented task allocation in human-machine collaboration system, in *2021 IEEE 4th Int. Conf. Inf. Syst. Comput. Aided Educ.* (IEEE, 2021), pp. 243–248.

[18] H. Gao, J. H. Feng, Y. Xiao, B. Zhang and W. D. Wang, A UAV-assisted multi-task allocation method for mobile crowd sensing, *IEEE Trans. Mob. Comput.* **22**(7) (2023) 3790–3804.

[19] T. X. Ma, P. Lu, F. Deng and K. Geng, Air–ground collaborative multi-target detection task assignment and path planning optimization, *Drones* **8**(3) (2024) 110.

[20] L. S. Liu, J. F. Lin, J. X. Yao, D. W. He, J. S. Zheng, J. Huang and P. Shi, Path planning for smart car based on Dijkstra algorithm and dynamic window approach, *Wirel. Commun. Mob. Comput.* **4** (2021) 1–12.

[21] X. Lai, J. H. Li and J. Chambers, Enhanced center constraint weighted A* algorithm for path planning of petrochemical inspection robot, *J. Intell. Robot. Syst.* **102**(4) (2021) 1–15.

[22] A. Vashisth, J. Ruckin, F. Magistri, C. Stachniss and M. Popovic, Deep reinforcement learning with dynamic graphs for adaptive informative path planning, *IEEE Robot. Autom. Lett.* **9**(9) (2024) 1–8.

[23] P. Venkatasivarambabu and R. Agrawal, Enhancing UAV navigation with dynamic programming and hybrid probabilistic route mapping: An improved dynamic window approach, *Int. J. Inf. Technol.* **16** (2024) 1023–1032.

[24] K. X. Cui, J. L. Song, L. Zhang, Y. Tao, W. Liu and D. Shi, Event-triggered deep reinforcement learning for dynamic task scheduling in multi-satellite resource allocation, *IEEE Trans. Aerosp. Electron. Syst.* **59**(4) (2023) 3766–3777.

[25] L. Lei, Y. Zhou and G. Yang, Multisource information fusion-based environment perception and dynamic model of underwater vehicle in irregular ocean environment, *Inf. Fusion.* **94** (2023) 257–271.

[26] T. Haesevoets, D. De Cremer, K. Dierckx and A. van Hiel, Human-machine collaboration in managerial decision making, *Comput. Hum. Behav.* **119** (2021) 106730.

[27] O. Mazhar, B. Navarro, S. Ramdani, R. Passama and A. Cherubini, A real-time human-robot interaction framework with robust background invariant hand gesture detection, *Robot. Comput. Integr. Manuf.* **60** (2019) 34–48.

[28] T. Becker and T. Schüle, Evaluating dynamic task scheduling with priorities and adaptive aging in a task-based runtime system, in *ARCS 2020: Architecture of Computing Systems*, eds. A. Brinkmann, W. Karl, S. Lankes, S. Tomforde, T. Pionteck and C. Trinitis (Springer, Cham, 2020), p. 12155.

[29] H. J. Huang, Y. C. Yang, H. Wang, Z. G. Ding, H. Sari and F. Adachi, Deep Reinforcement Learning for UAV Navigation Through Massive MIMO Technique, *IEEE Trans. Veh. Technol.* **69**(1) (2020) 1117–1121.

[30] J. N. Yasin, S. A. S. Mohamed, M.-H. Haghbayan, J. Heikkonen, H. Tenhunen and J. Plosila, Unmanned aerial vehicles (UAVs): Collision

avoidance systems and approaches, *IEEE Access* **8** (2020) 105139–105155.

[31] G. Lalitha, B. Rajalakshmi, M. Gupta, Praveen, V. Kumar and A. A. Alwan, Reinforcement learning for real-time decision-making in autonomous air combat systems, in *2023 10th IEEE Uttar Pradesh Section Int. Conf. Electr. Electronics Comput. Eng.* (IEEE, India, 2023), pp. 1593–1598.

[32] Z. Zhang, J. Jiang, H. Y. Xu and W. A. Zhang, Distributed dynamic task allocation for unmanned aerial vehicle swarm systems: A networked evolutionary game-theoretic approach, *Chin. J. Aeronaut.* **37**(6) (2024) 182–204.

[33] B. X. Zhao, H. B. Dong, Y. J. Wang and T. W. Pan, A task allocation algorithm based on reinforcement learning in spatio-temporal crowdsourcing, *Appl. Intell.* **53** (2023) 13452–13469.

[34] Z. R. Huang, C. C. Chen and M. Pan, Multiobjective UAV path planning for emergency information collection and transmission, *IEEE Internet Things J.* **7**(8) (2020) 6993–7009.

[35] Z. C. Hu, H. Chen, E. Lyons, S. Solak and M. Zink, Towards sustainable UAV operations: Balancing economic optimization with environmental and social considerations in path planning, *Transport. Res. E-log.* **181** (2024) 103314.

[36] P. Kannisto, D. Hästbacka and A. Marttinen, Information exchange architecture for collaborative industrial ecosystem, *Inf. Syst. Front.* **22** (2020) 655–670.

[37] İ. Güven and E. Yanmaz, Multi-objective path planning for multi-UAV connectivity and area coverage, *Ad Hoc Netw.* **160** (2024) 103520.

[38] X. J. Liu, X. W. Wang, J. Jia, J. H. Lv and N. Bartolini, Distributed deployment in UAV-assisted networks for a long-lasting communication coverage, *IEEE Syst. J.* **16**(3) (2022) 4130–4138.

[39] R. X. Sun, D. Zhao, L. G. Ding, J. Zhang and H. D. Ma, UAV-Net+: Effective and energy-efficient UAV network deployment for extending cell tower coverage with dynamic demands, *IEEE Trans. Veh. Technol.* **72**(1) (2023) 973–985.

[40] D. Liu, L. Dou, R. Zhang, X. Zhang and Q. Zong, Multi-agent reinforcement learning-based coordinated dynamic task allocation for heterogenous UAVs, *IEEE Trans. Veh. Technol.* **72**(4) (2023) 4372–4383.

[41] P. F. Wan, S. K. Wang, G. Y. Xu, Y. J. Long and R. Q. Hu, Hybrid heuristic-based multi-UAV route planning for time-dependent data collection, *IEEE Internet Things J.* **11**(13) (2024) 24134–24147.

[42] T. Rashid, G. Farquhar, B. Peng and S. Whiteson, Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning, *Adv. Neural Inf. Process. Syst.* **33** (2020) 10199–10210.

[43] D. X. Wei, L. Zhang, Q. Liu, H. Chen and J. Huang, UAV swarm cooperative dynamic target search: A MAPPO-based discrete optimal control method, *Drones* **8**(6) (2024) 214.

[44] E. Y. Zhao, N. Zhou, C. J. Liu, H. F. Su, Y. Liu and J. M. Cong, Time-aware MADDPG with LSTM for multi-agent obstacle avoidance: A comparative study, *Complex Intell. Syst.* **10** (2024) 4141–4155.