frontiers | Frontiers in Communications and Networks

TYPE Original Research
PUBLISHED 10 June 2024
DOI 10.3389/frcmn.2024.1390909

OPEN ACCESS

EDITED BY
Oluwakayode Onireti,
University of Glasgow, United Kingdom

REVIEWED BY
Kosta Dakic,
The University of Sydney, Australia
Attai Ibrahim Abubakar,
University of Glasgow, United Kingdom

*CORRESPONDENCE
Chao Yang,
✉ chyang513@gdut.edu.cn

RECEIVED 24 February 2024
ACCEPTED 13 May 2024
PUBLISHED 10 June 2024

CITATION
Lai Q, Xie R, Yang Z, Wu G, Hong Z and Yang C (2024), Efficient multiple unmanned aerial vehicle-assisted data collection strategy in power infrastructure construction.
Front. Comms. Net 5:1390909.
doi: 10.3389/frcmn.2024.1390909

Efficient multiple unmanned aerial vehicle-assisted data collection strategy in power infrastructure construction

Qijie Lai[1], Rongchang Xie[1], Zhifei Yang[1], Guibin Wu[2], Zechao Hong[2] and Chao Yang[3]*

[1]Department of Infrastructure, Guangdong Power Grid Corporation, Guangzhou, China, [2]Yunfu Power Supply Bureau, Guangdong Power Grid Corporation, Yunfu, China, [3]School of Automation, Guangdong University of Technology, Guangzhou, China

Efficient data collection and sharing play a crucial role in power infrastructure construction. However, in an outdoor remote area, the data collection efficiency is reduced because of the sparse distribution of base stations (BSs). Unmanned aerial vehicles (UAVs) can perform as flying BSs for mobility and line-of-sight transmission features. In this paper, we propose a multiple temporary UAV-assisted data collection system in the power infrastructure scenario, where multiple temporary UAVs are employed to perform as relay or edge computing nodes. To improve the system performance, the task processing model selection, communication resource allocation, UAV selection, and task migration are jointly optimized. We designed a QMIX-based multi-agent deep reinforcement learning algorithm to find the final optimal solutions. The simulation results show that the proposed algorithm has better convergence and lower system costs than the current existing algorithms.

KEYWORDS

data collection, unmanned aerial vehicle, task migration, multi-agent deep reinforcement learning, QMIX

1 Introduction

With the rapid economic development, more electric power infrastructures have been built rapidly, which causes more cases of outdoor power line erection, distribution station construction, and other types of facilities (Kumar et al., 2017). The South China Power Grid, along with other companies, has initiated comprehensive power infrastructure management measures. They utilize advancements from the Internet of Things (IoT), communication networks, and artificial intelligence technologies to oversee the power infrastructure processes to enhance the safety protocols for the construction workers and maintain the quality standards of the projects rigorously. The holistic strategy marks significant progress in integrating modern technology with traditional power systems, reflecting an evolving landscape in infrastructure management. Along with the process, a large amount of onsite data need to be collected and fused for processing (L'Abbate, 2023; Stoupis et al., 2023). For example, in order to construct a digital twin (DT) for outdoor power infrastructure construction systems, the ground nodes with sensing devices need to share information with other nodes continuously. In particular, the shared information is not only associated with the ground nodes but also with the environment and the construction workers. However, the various types of data to be processed and the complex construction environment make the data collection of outdoor power infrastructure extremely difficult.

Due to the superior mobility, ease of deployment, and direct line-of-sight (LoS) communication advantages, unmanned aerial vehicles (UAVs) are increasingly utilized as relay or edge computing nodes in data collection networks (Jayakody et al., 2020; Rahman et al., 2022), specifically in power infrastructure contexts. Notably, in addition to those that are strategically deployed, there exist multiple UAVs undertaking separate missions, such as package delivery, target tracking, and environmental monitoring, often operating in areas encompassing the power infrastructure facilities. A significant portion of the computational and communication resources on these incidental UAVs are typically idle. Engaging these UAVs in infrastructure monitoring initiatives could profoundly enhance the network communicative efficiency and computational capacity. However, it is critical that these UAVs are primarily dedicated to their original assignments; their operations are designed and hard to be changed.

Currently, in the UAV-assisted IoT, UAVs function as relay nodes (Rahman et al., 2022; Jayakody et al., 2020; Zhang et al., 2018) or edge computing nodes (Jeong et al., 2018; Liu et al., 2020) primarily, with particular emphasis on their flight trajectories, energy consumption, and task offloading strategies. For instance, Jeong et al. (2018) optimized UAV communication resource allocation and trajectory planning to minimize the overall system energy consumption, ensuring service quality. Conversely, Liu et al. (2020) introduced an edge computing framework utilizing a fleet of UAVs as relay nodes. The task offloading and UAV flight planning are optimized jointly to meet the users' computational demands and latency sensitivities. Addressing the issue of base station (BS) edge servers in power infrastructure networks occasionally failing to satisfy the users' computational needs, Hu et al. (2019) and Peng et al. (2020) advocated for deploying UAVs equipped with edge computing servers to overwhelmed areas; the proposed strategy can mitigate resource scarcity in roadside units during peak periods effectively. Zhang et al. (2022) established a UAV-assisted edge vehicular network computing structure with energy harvesting. UAVs aid vehicles in executing onboard task computations. Employing wireless power transfer (WPT) technology, UAVs harvest energy from base stations and vehicles. The UAV speed, computation, and communication resource allocation are jointly optimized to enhance UAV data-processing efficiency. Ng et al. (2020) utilized UAVs as relay nodes to improve the efficiency of the interactions between Internet of Vehicles (IoV) components and the federated learning (FL) server in the IoV with an FL edge computing server, thereby enhancing FL precision. However, the primary objective is to escalate UAV profitability, and the results indicate that the system stability also needs be enhanced. Furthermore, Liu et al. (2022) proposed the strategic deployment of UAVs within edge vehicular networking contexts, aiming to cater to network-connected autonomous vehicles that exhibit intensive computational and communication resource requests; the allocation of UAV communication and computational resources is optimized to maximize the system accessible resource capacity. Sun et al. (2023b) optimized the schedules among UAVs to ensure the shortest response time in the UAVs with edge computing services that cover the IoV users for data collection. Moreover, the communication bandwidth allocation and flying trajectory are jointly optimized by Wang et al. (2023b). Lyu et al. (2019) and Peng et al. (2020) confronted the disparity between fluctuating vehicle densities and the static establishment of roadside unit (RSU), and the flexible UAV deployment strategy is proposed to equilibrate network Quality of service (QoS) and UAV deployment frequency.

However, in power infrastructure scenarios, the ground nodes (such as wind generator and wire tower) have several types of tasks that need to be processed. Because the UAV can function as both the relay and edge computing node, it can select a suitable task processing model based on the mobility of the UAV and the task computing resource requests. The UAV can select a suitable task processing model. In addition, when a single UAV acts as a relay node or edge computing node, its resources are not sufficient, and it is difficult to handle intensive task processing requests in complex outdoor scenarios. A UAV swarm composed of multiple UAVs can solve this problem effectively. Compared with the single-UAV scenario, the design of the multiple UAV-assisted edge network optimization strategy also needs to consider the problems of relay selection and task migration. Thus, it is necessary to select the appropriate node in the data uploading phase and the selection of the task migration node when the ground node is moving away from the communication range of the UAV, where the current task is located. In this paper, we propose a UAV selection, communication resource allocation, and task migration joint optimal strategy based on a multi-agent deep reinforcement learning algorithm via QMIX (Rashid et al., 2018), and the main optimization objective is to minimize the cost of the ground users under the task completion delay constraint. The main contributions of this paper are summarized as follows:

1) We construct a temporary UAV-assisted data collection system model in the power infrastructure scenario, and multiple UAVs are employed to function as relay or edge computing nodes.
2) We formulate a joint optimization method to minimize the system cost, where the task processing model, communication resource allocation, UAV selection, and task migration are jointly optimized.
3) We present a QMIX-based multi-agent deep reinforcement learning algorithm to find the final optimal solutions.

Moreover, several simulation examples are presented to show that the proposed UAV-assisted data collection method can reduce the task-completion delay and the system cost. The rest of this paper is organized as follows: the system model is proposed in Section 2; Section 3 describes the joint problem solution via the QMIX-based algorithm; Section 4 presents multiple simulations and discusses the simulation results; and finally, the conclusion of the paper is presented in Section 5.

2 System model

Figure 1 illustrates the system model of the multiple temporary UAV-assisted data collection network. We consider a power infrastructure scenario in the wild, as shown in Figure 1; several ground nodes (including wind generator and wire tower) have a high demand for task transmission to build a DT system for the

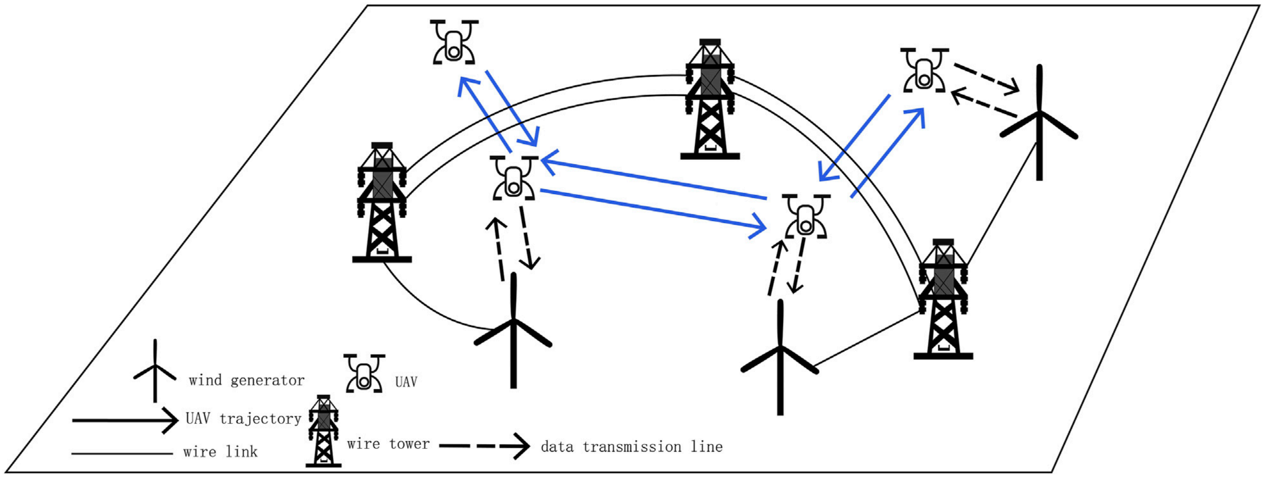

FIGURE 1
Multi-temporary unmanned aerial vehicle (UAV)-assisted data collection system model.

TABLE 1 Main symbols.

Notation	Description
i	Index of volunteer nodes
j	Index of unmanned aerial vehicle (UAV)
$h_{ij}(t)$	Communication channel gain between UAV j and node i at time t
S_i	Size of task i
β_0	Channel power per unit distance
p_{iu}	Transmission power of a volunteer node
σ^2	Gaussian white noise
B_0	Sub-channel bandwidth
C_T	Data transmission unit price
C_P	Data processing unit price
$\boldsymbol{O}_i$	Processing result size
$\boldsymbol{C}_i$	Computing resource requirement
$T_{u,i}$	Time of successful task uploading
$T'_{p,i}$	Total processing time of task i

reconstruction of the infrastructure scenario to improve the management quality. Multiple temporary UAVs pass the power gird construction site, and the temporary passing UAVs have their own missions; the available resources of a single UAV are not enough, and multiple UAVs can cooperative to finish the data collection efficiently. They form a UAV network and can function as both the relay and edge computing nodes. To obtain enough information to improve the advanced decision-making system accuracy, the ground nodes in the infrastructure scenario need to continuously share information with other nodes through the edge network, and the tasks are collected by relying on their own sensing devices. The information is not only associated with the ground nodes but also with the environment and the construction workers. The decision-making center cannot change the flying trajectory and other parameters, and we set that the ground nodes should pay when they use the UAVs for data collection. The energy consumption and available varying computation resources are not considered in this paper. The main symbols used are summarized in Table 1.

When a task node collects information, it first sends out an information task via the common orthogonal frequency-division multiplexing access (OFDMA) technology (Xia et al., 2021), and then, a volunteer node receives the task. After receiving the task, the volunteer node immediately packages its own collected real-time information and sends it directly back to the task node via wireless communication or to the UAV, which then acts as a relay node to forward the data to the task node. For clarity in the discussion, we assign numbers to both the task nodes and volunteer nodes. The task node is numbered as 0, while the volunteer nodes are designated as i, with i belonging to the set $i \in \boldsymbol{I} = \{1, 2, \ldots, I\}$. Similarly, the UAV index is denoted by j, and $j \in \boldsymbol{J} = \{1, 2, \ldots, J\}$. In addition, using the UAV as a relay node to collect information is the main scenario. The data transmission model for collecting information by the task nodes is shown in Figure 2.

In the single-UAV scenario, there are no idle UAV nodes available for selection, eliminating the need for volunteer nodes to make relay choices when uploading data. However, with multiple UAVs, volunteer nodes must identify and select an appropriate UAV as a relay node before data transmission to ensure the completion of the data collection task as effectively as possible. The UAV is denoted by u_i, representing the relay UAV number chosen for task i. Concurrently, $\boldsymbol{I}_j$ denotes the set of volunteer vehicles that have selected UAV j as their relay node. As shown in Figure 2, after finalizing their relay selection, the volunteer nodes initiate the data uploading process to the corresponding UAVs through wireless links. For tasks that have designated the same UAV as a relay node, the UAV has the capability to process the tasks in a parallel manner. Following the completion of these uploads, immediate commencement of task processing occurs, affecting the system performance. Due to the uncontrolled flight paths of the

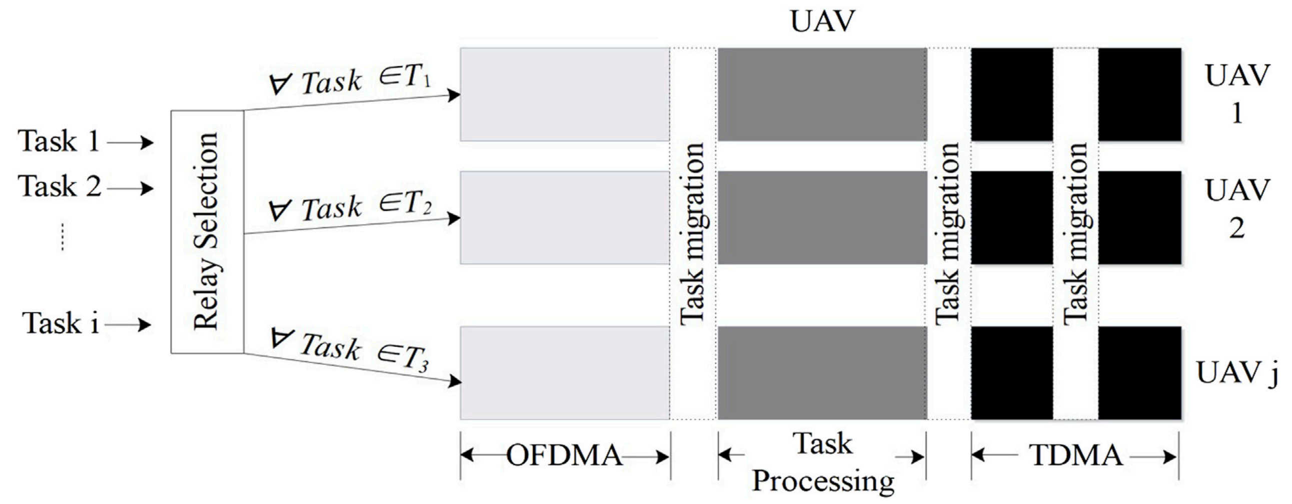

FIGURE 2
Multi-temporary UAV data transmission model.

TABLE 2 Task migration process in different situations.

Situations	Specific migration action
Situation 1: no migration needed	No migration action
Situation 2: occurs during the task uploading phase	Migrates the uploaded data to the new temporary unmanned aerial vehicle (UAV) when the upload is completed
Situation 3: occurs during the task processing phase	Migrates the results to the new temporary UAV when data processing is completed
Situation 4: occurs during the download phase	Transfers the remaining processing results to the new temporary UAV

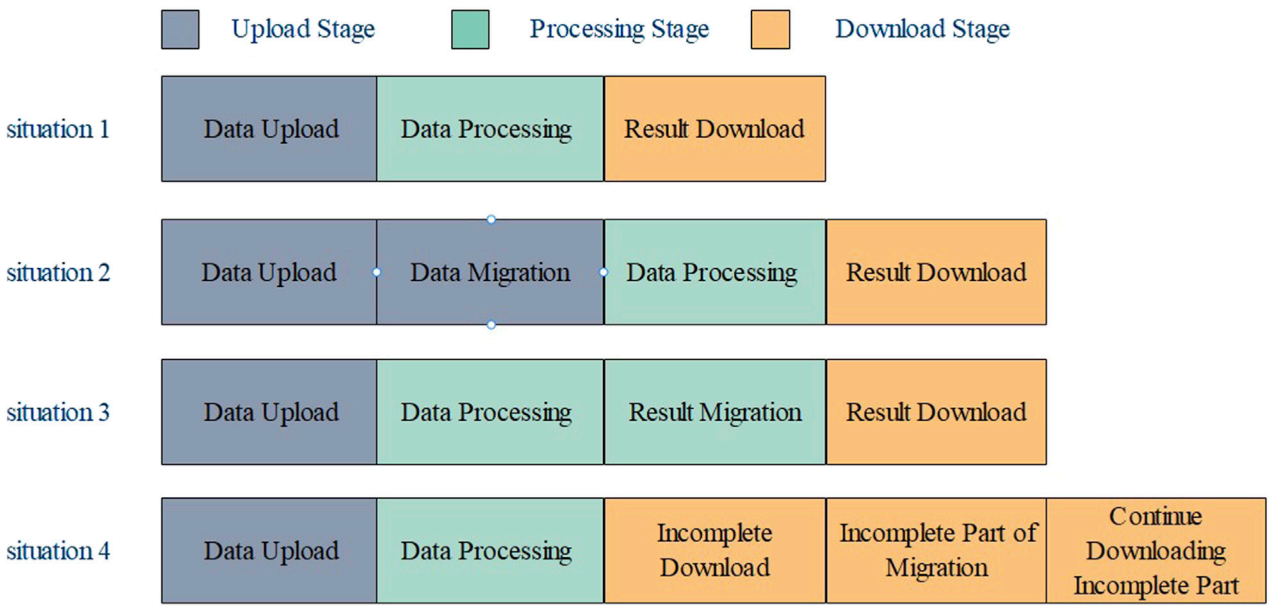

FIGURE 3
Task migration in different situations.

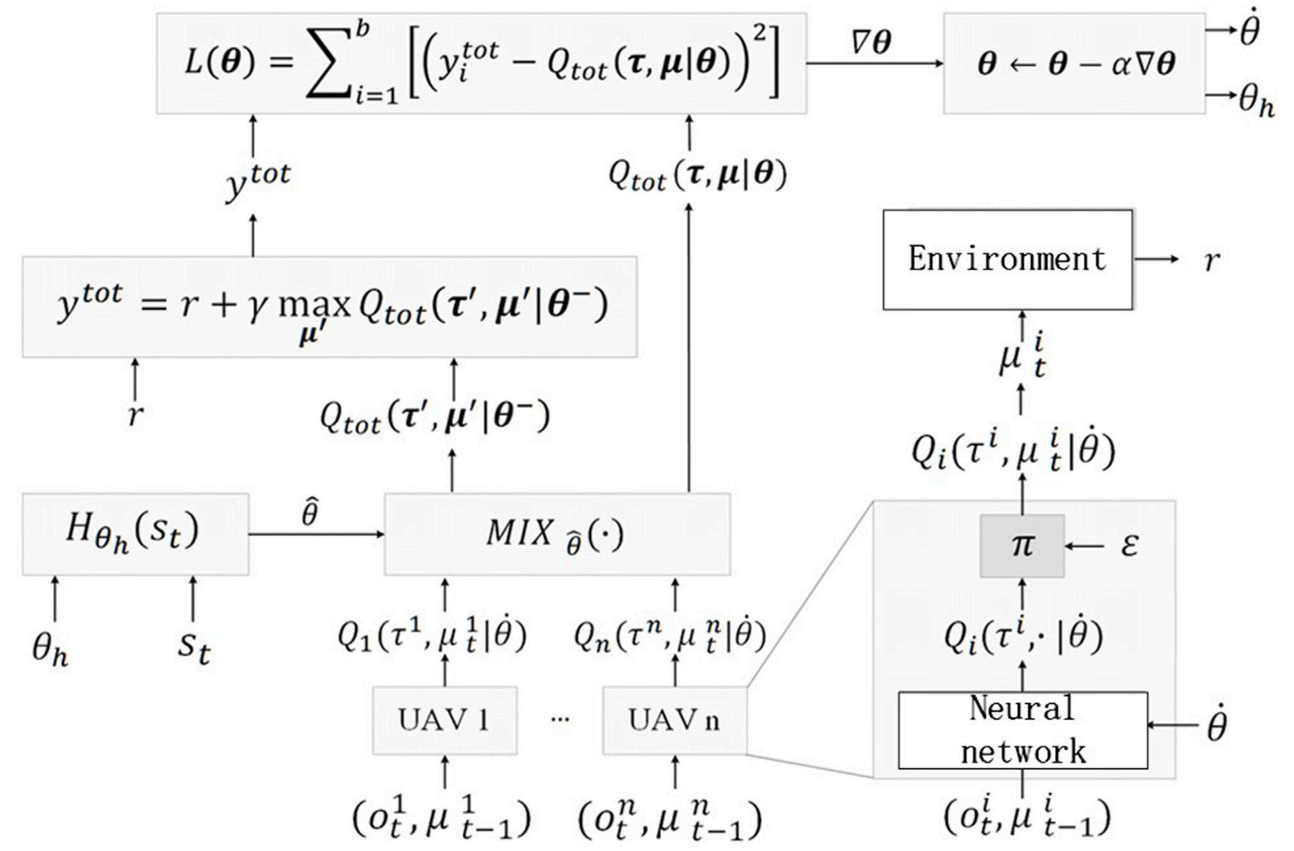
$L(\boldsymbol{\theta}) = \sum_{i=1}^{b}\left[\left(y_i^{tot} - Q_{tot}(\boldsymbol{\tau},\boldsymbol{\mu}|\boldsymbol{\theta})\right)^2\right]$
$\nabla\boldsymbol{\theta}$
$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha\nabla\boldsymbol{\theta}$
$\dot{\theta}$
θ_h
y^{tot}
$Q_{tot}(\boldsymbol{\tau},\boldsymbol{\mu}|\boldsymbol{\theta})$
Environment
r
$y^{tot} = r + \gamma \max_{\boldsymbol{\mu}'} Q_{tot}(\boldsymbol{\tau}',\boldsymbol{\mu}'|\boldsymbol{\theta}^-)$
μ_t^i
r
$Q_{tot}(\boldsymbol{\tau}',\boldsymbol{\mu}'|\boldsymbol{\theta}^-)$
$Q_i(\tau^i,\mu_t^i|\dot{\theta})$
$H_{\theta_h}(s_t)$
$\hat{\theta}$
$MIX_{\hat{\theta}}(\cdot)$
π
ε
θ_h
s_t
$Q_1(\tau^1,\mu_t^1|\dot{\theta})$
$Q_n(\tau^n,\mu_t^n|\dot{\theta})$
$Q_i(\tau^i,\cdot|\dot{\theta})$
UAV 1
...
UAV n
Neural network
$\dot{\theta}$
(o_t^1,μ_{t-1}^1)
(o_t^n,μ_{t-1}^n)
(o_t^i,μ_{t-1}^i)

FIGURE 4
Framework of the QMIX-based selection strategy on relay and migration target.

TABLE 3 Parameters.

Parameter	Description	Value
β_0	Channel power per unit distance	50 dBm
p_{iu}	Transmission power of a volunteer node	10 dBm
σ^2	Gaussian white noise	−30 dBm
B_0	Sub-channel bandwidth	1 M
C_T	Data transmission unit price	0.2
C_P	Data processing unit price	0.5
$\boldsymbol{O}_i$	Processing result size	$\{S_i, 0.7S_i, 0.01S_i\}$
$\boldsymbol{C}_i$	Computing resource requirement	$\{0, S_i, 2S_i\}$
θ_h	Initialized hypernetwork parameters	[2, 10]

temporary UAVs and the mobility of ground users, the task node might exit the communication range of a UAV before task completion. Under these scenarios, if the task is still valid and not yet completed, a process called task migration is initiated. It transfers the task to a UAV corresponding to a neighboring node, thereby ensuring the continuation and subsequent task completion. The efficiency and reliability of tasks amidst dynamic operational environments can thus be ensured. During the task migration process, it is essential to select a target UAV first. The selected target UAV for task $\boldsymbol{i}$ is denoted as g_i, where g_i is a member of the set $\boldsymbol{G}$ and $g_i \in \boldsymbol{G} = \boldsymbol{J} \cup \{0\}$. Considering the task consumes more communication resources and time after migration, i.e., the cost of the task will increase while the success rate of the task will decrease, it is necessary to optimize the migrating selection to avoid possible greater losses. $g_i = 0$ signifies the direct abandonment of the task, reflecting a conservative option under consideration.

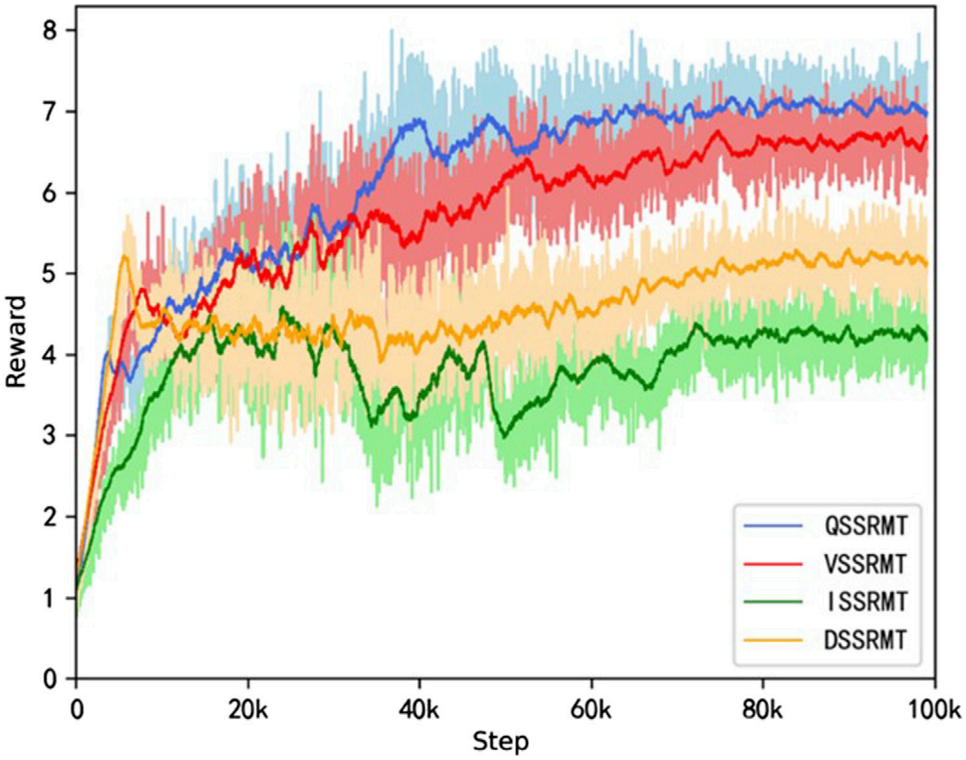

FIGURE 5
Reward curve of different algorithms.

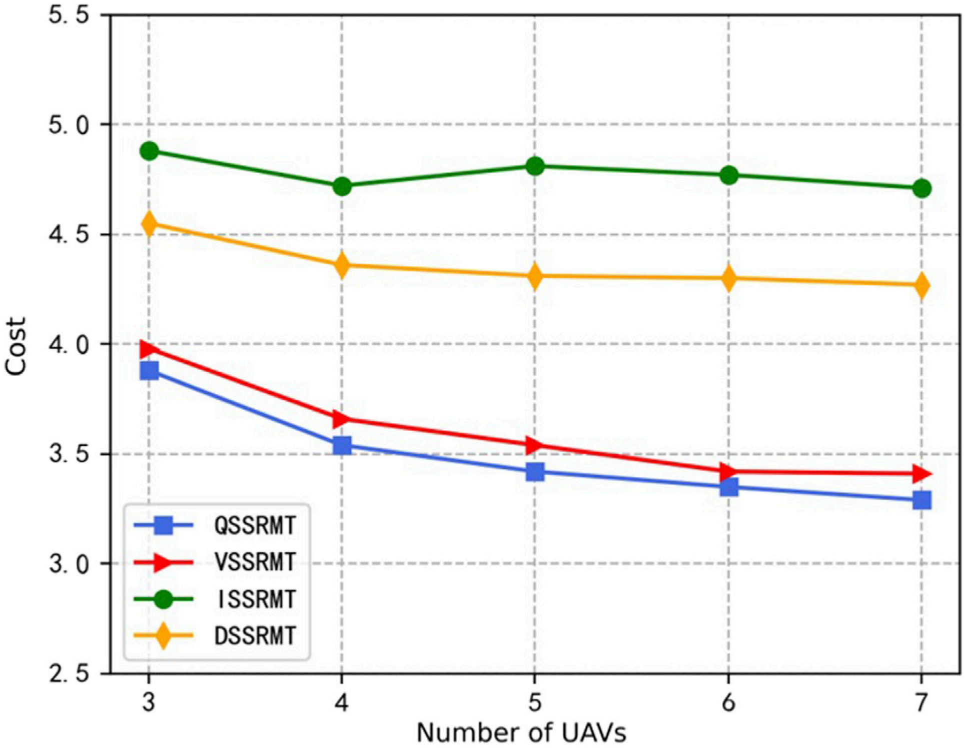

FIGURE 6
Relationship between the cost and number of UAVs.

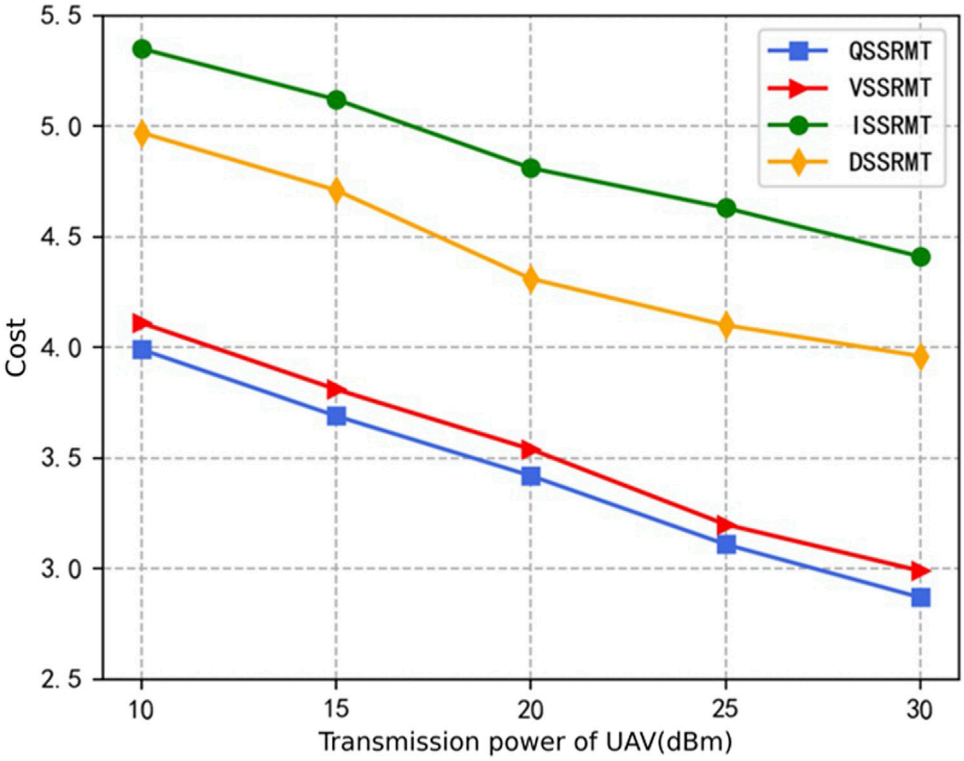

FIGURE 7
Relationship between the cost and transmission power of UAVs.

Moreover, the task node departure from the UAV communication coverage can transpire any moment during the data transmission phase. The specific content should be migrated, and it depends on the exact timing; the detailed scenarios are outlined in Table 2. This variability necessitates a dynamic approach to task management, ensuring that task transitions are handled efficiently based on the circumstances of the network connectivity and the task progress.

2.1 Delay calculation

2.1.1 Scenarios with no migration needed

UAVs can function as aerial platforms, establishing LoS links with ground nodes; thus, we utilize the free space path loss (FSPL) model to characterize the communication channel between the UAVs and the ground nodes (Hossein Motlagh et al., 2016). The communication channel gain between UAV j and node i at time t is denoted as $h_{ij}(t)$, which can be expressed as follows:

$$h_{ij}(t) = \beta_0 d_{ij}^{-2}(t),$$

where β_0 is the channel power gain per unit distance and d_{ij} denotes the straight-line distance between the UAV j and the node i. The channel bandwidth is $B = Kb$, i.e., the bandwidth B is divided into K copies of the sub-channel with bandwidth b. The match between the sub-channels and the volunteer node is the resource allocation problem. n_{ij} denotes the number of sub-channels obtained by node i when uploading data to UAV j, where $i \in I_j$ and $\sum_{i \in I_j} n_{ij} = K$. The upload link channel capacity from the volunteer node i to the UAV j at time t is as follows:

$$R_{ij}(t) = n_{ij} b \log_2\left(1 + \frac{p_{ij} h_{ij}(t)}{\sum_{k \neq i} p_{kj} h_{kj}(t) + \sigma^2}\right),$$

where p_{ij} denotes the signal transmit power of the volunteer node i; $n_{ij}b$ denotes the channel bandwidth for uploading data. $\sum_{k \neq i} P_{kj} h_{kj}(t)$ indicates an interfering signal and σ^2 denotes Gaussian white noise. The size of task i is represented by S_i, which is calculated as follows:

$$S_i = \int_{t_{su}}^{t_{eu}} R_{ij}(t) dt,$$

where t_{su} and t_{eu} denote the upload starting and ending time, respectively. Denoting the preparation time of task i by T_i^g, the time consumed for successful task uploading is as follows:

$$T_{u,i} = T_i^g + t_{eu} - t_{su}. \tag{1}$$

After the data processing phase, the processing results are sent to the task node immediately. We consider the Time Division Multiple Address (TDMA) technology so that the task in the current transmission has all the available channels currently. The data download rate can be expressed as follows:

$$R_{ji}(t) = B \log_2\left(1 + \frac{p_{ji} h_{ji}(t)}{\sum_{k \neq j} p_{ki} h_{ki}(t) + \sigma^2}\right). \tag{2}$$

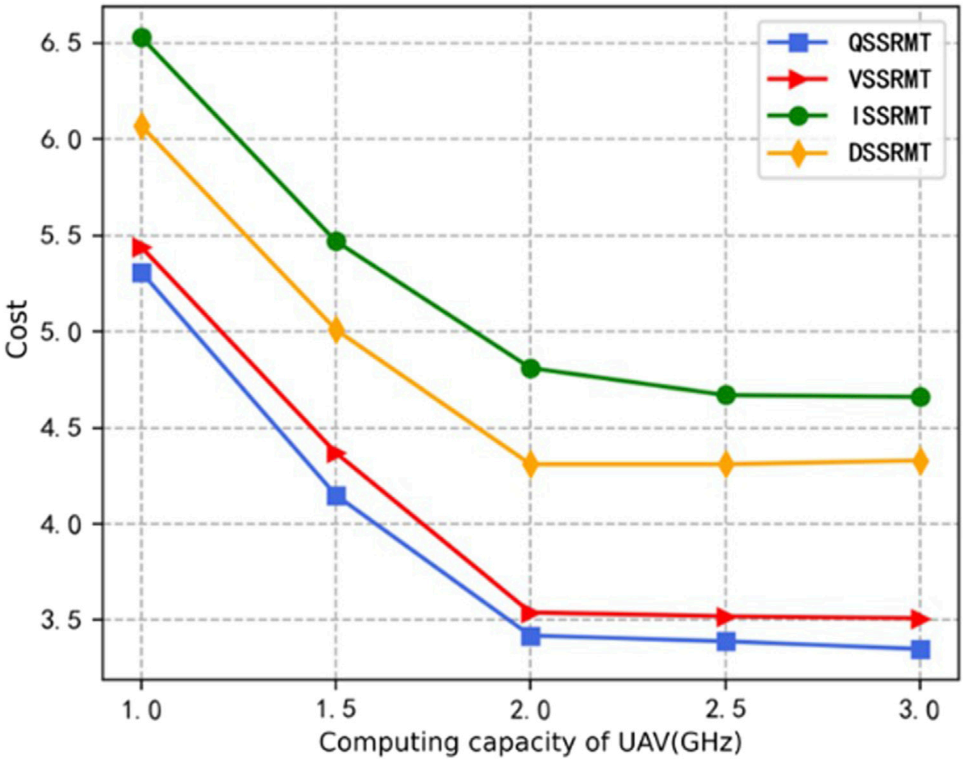

FIGURE 8
Relationship between the cost and UAV computing capacity.

The size of the processed results after the completion of task i is denoted as o_i, which is as follows:

$$o_i = \int_{t_{sd}}^{t_{ed}} R_{ji}(t)dt,$$

where t_{sd} and t_{ed} denote the download starting time and ending time, respectively. The time consumption in the download phase $T_{download,i}$ is denoted as follows:

$$T_{d,i} = t_{ed} - t_{sd}.$$

2.1.2 Scenarios containing task migration

Task migration is accomplished through communication links between the UAVs (Huang et al., 2021), for which the channel rate is expressed as follows:

$$R_{jg}(t) = B\log_2\left(1 + \frac{p_{jg}h_{jg}(t)}{\sum_{k\neq j} p_{kg}h_{kg}(t) + \sigma^2}\right).$$

As shown in Figure 3, task migration can occur at any stage. For scenario 2, where the task migration occurs during the task uploading phase, the starting and ending times for migration are set as t_{sm} and t_{em}, respectively. Consequently, the total time consumption for the data-uploading phase of task i is represented as follows:

$$T'_{u,i} = T_{u,i} + T_{m,i} = T_{u,i} + t_{em} - t_{sm}.$$

In addition, t_{sm} and t_{em} satisfy the following equation:

$$S_i = \int_{t_{sm}}^{t_{em}} R_{ij}(t)dt.$$

For situation 3, the total processing time of task i is as follows:

$$T'_{p,i} = T_{p,i} + T_{m,i} = T_{p,i} + t_{em} - t_{sm}.$$

In addition, t_{sm} and t_{em} satisfy the following equation:

$$o_i = \int_{t_{sm}}^{t_{em}} R_{jg}(t)dt,$$

where $T_{p,i}$ is the time consumed for the complete data processing.

For scenario 4, the total time consumed during the result download phase, denoted as $T'_{d,i}$ comprises three segments: the duration of the initial data download $T^1_{d,i}$, the time taken for the migration of the remaining data $T_{m,i}$, and the time required to download the remaining data $T^2_{d,i}$. It is assumed that the starting time of the first download is t_{sd}, the ending time of the remaining data download is t_{ed}, the starting time and ending time of task migration are t_{sm} and t_{em}, respectively, and the size of the remaining data is o'_i; then,

$$T'_{d,i} = t_{ed} - t_{sd},$$

where t_{sd} and t_{sm} satisfy the following equation:

$$\left(o_i - o'_i\right) = \int_{t_{sd}}^{t_{sm}} R_{ij}(t)dt.$$

In addition, t_{sm}, t_{em}, and t_{ed} satisfy the following equation:

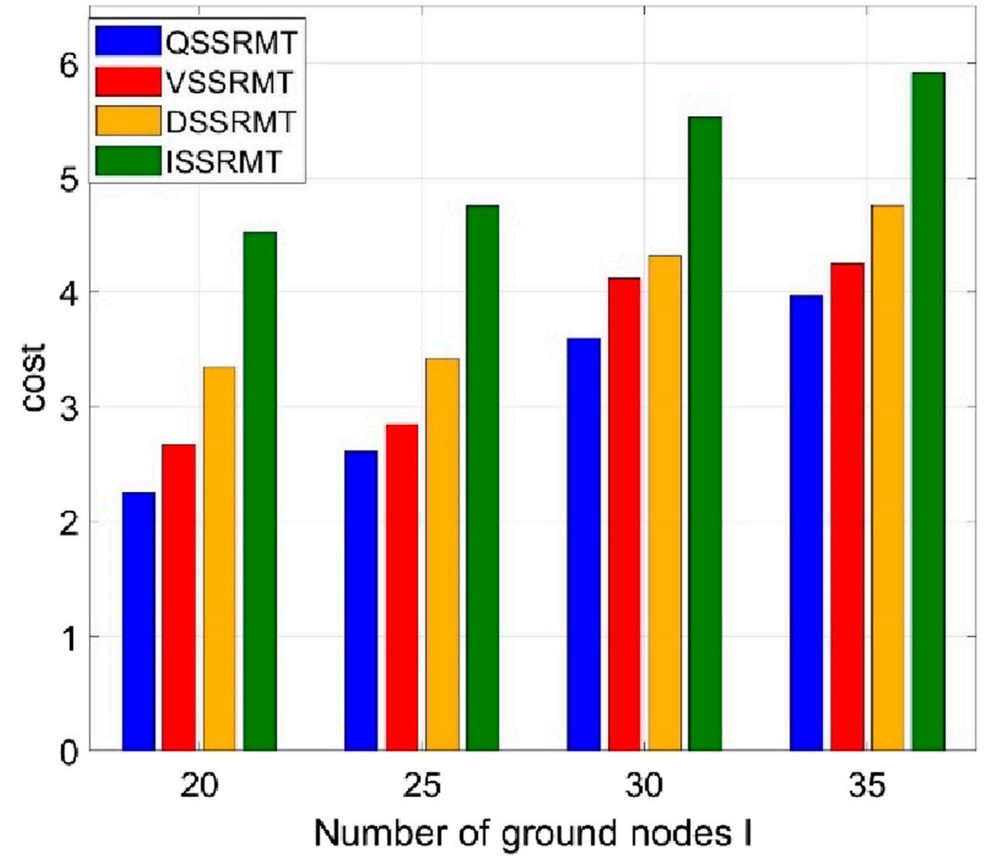

FIGURE 9
Relationship between the cost and number of ground nodes I.

$$o_i' = \int_{t_{sm}}^{t_{em}} R_{jg}(t)dt = \int_{t_{em}}^{t_{ed}} R_{g0}(t)dt,$$

where $R_{g0}(t)$ is the download link communication rate between the new temporary UAV g and the mission vehicle 0.

The total time $T_{tot,i}$ spent in completing task i can be described as follows:

$$T_{tot,i} = \begin{cases} T_{u,i} + T_{p,i} + T_{d,i}, \text{case 1} \\ T_{u,i}' + T_{p,i} + T_{d,i}, \text{case 2} \\ T_{u,i} + T_{p,i}' + T_{d,i}, \text{case 3} \\ T_{u,i} + T_{p,i} + T_{d,i}', \text{case 4} \end{cases},$$

where $\boldsymbol{m_i} = (m_{i,1}, m_{i,2}, m_{i,3})$ is used to indicate whether task i has migrated or not, $m_{i,1}, m_{i,2}, m_{i,3} \in \{0, 1\}$, $m_{i,1} = 1$ indicates that the migration occurs in the data uploading phase, $m_{i,2} = 1$ indicates that it occurs in the data processing phase, and $m_{i,3} = 1$ indicates that it occurs in the data downloading phase. Considering the impact of task migration on the efficiency of data collection, and to simplify the model, we restrict the task migration behavior to occur only once (i.e., $\|\boldsymbol{m}_i\|_1 \le 1$). The value of $\boldsymbol{m}_i$ depends on the moment of the occurrence of task migration, t_{sm}, and the migration target, g_i. Thus, $T_{tot,i}$ can be rewritten as follows:

$$T_{tot,i} = (1 - m_{i,1}, m_{i,1}) \cdot (T_{u,i}, T_{u,i}') + (1 - m_{i,2}, m_{i,2}) \cdot (T_{p,i}, T_{p,i}') + (1 - m_{i,3}, m_{i,3}) \cdot (T_{d,i}, T_{d,i}').$$

The time taken to complete all the tasks in a cycle is as follows:

$$T_{tot} = \max_{j, j\in \boldsymbol{J}} \left(\max_{i, i\in \boldsymbol{I}_j} \left\{ (1 - m_{i,1}, m_{i,1}) \cdot (T_{u,i}, T_{u,i}') \right\} + \sum\nolimits_{i\in \boldsymbol{I}_j} (1 - m_{i,2}, m_{i,2}) \cdot (T_{p,i}, T_{p,i}') + \sum\nolimits_{i\in \boldsymbol{I}_j} (1 - m_{i,3}, m_{i,3}) \cdot (T_{d,i}, T_{d,i}') \right).$$

2.2 Cost calculation

The total system costs are divided into communication costs and computational costs. P_T represents the unit price of data transmission (the cost of transmitting 1 Mb of data), and P_P represents the unit price of computational resources (the cost for consuming 1 G CPU cycles). c_i denotes the computational resources consumed by task i. Under typical circumstances, the cost required for task nodes to successfully complete data task i can be expressed as follows:

$$cost_i = P_T(S_i + o_i) + P_P c_i.$$

When task migration is included, the cost also includes the expenses incurred for task migration. The main costs incurred during task migration are communication costs, which can be expressed as follows:

$$cost_{m,i} = P_T\left(m_{i,1}S_i + m_{i,2}o_i + m_{i,3}o_i^{'}\right).$$

Then, $cost_i$ can be expressed as follows:

$$cost_i = P_T(S_i + o_i) + P_P c_i + P_T\left(m_{i,1}S_i + m_{i,2}o_i + m_{i,3}o_i^{'}\right).$$

It is important to highlight that $cost_i$ accounts for the cost associated with a successful task, but not every task necessarily reaches completion. Tasks are often highly time-sensitive, which means that they might be terminated at any stage due to exceeding the time constraints. To uniformly describe the costs related to both successful and unsuccessful tasks, we employ coefficients ξ_u, ξ_p, ξ_d and ξ_m. These coefficients represent the progress in data upload, data processing, processing result download, and task migration throughout the data collection process, respectively. With binding coefficients ξ_u, ξ_p, ξ_d, and ξ_m, the total cost consumed by the task i can be re-expressed as follows:

$$cost_i = P_T(\xi_u S_i + \xi_d o_i) + P_P \xi_p c_i + P_T \xi_m\left(m_{i,1}S_i + m_{i,2}o_i + m_{i,3}o_i^{'}\right).$$

Among these, $\xi_u, \xi_p, \xi_d, \xi_m \in [0,1]$. It should be noted that the $\xi_u, \xi_p, \xi_d, \xi_m$ corresponding process has a time-sequence relationship, so the value of the coefficient corresponding to the process earlier in time will affect the coefficient corresponding to the process later in time. For example, when $\xi_u < 1$, the uploading phase is not completed; there must be $\xi_p = 0$, i.e., the processing phase cannot be started.

2.3 Optimized problem description

According to the description of the above system model, we establish the joint optimization problem. Under the premise of satisfying the task delay constraints, channel resource constraints, and decision space constraints, the relay selection decision variable u_i, the number of channel allocations n_{ij}, the task processing decision variable $\boldsymbol{\alpha}_i$, and the task migration decision variable g_i are jointly optimized to minimize the cost incurred by the users to complete the information collection task in the multi-UAV-empowered power infrastructure data collection network system. The optimization problem **P1** is expressed as follows:

P1: $\min_{\{u_i, n_{ij}, \boldsymbol{\alpha}_i, g_i\}} \sum_{i\in \boldsymbol{I}} cost_i$

Subject to:

C1: $T_{tot,i} \le T_{req,i}, \forall i \in \boldsymbol{I}$

C2: $T_{tot} \le T_{req}^{th}$

C3: $u_i \in \boldsymbol{J}, \forall i \in \boldsymbol{I}$

C4: $\sum_{i\in \boldsymbol{I}_j} n_{ij} = K$

C5: $\|\boldsymbol{\alpha}_i\|_1 = 1$

C6: $n_{ij} \in \{1, 2, \ldots, K\}, \forall i \in \boldsymbol{I}_j,\ g_i \in \boldsymbol{G} = \boldsymbol{J} \cup \{0\}, \forall i \in \boldsymbol{I}$

C7: $\bigcap_{j\in \boldsymbol{J}} \boldsymbol{I}_j = \varnothing,\ \bigcup_{j\in \boldsymbol{J}} \boldsymbol{I}_j = \boldsymbol{I}$

where T_{req}^{th} represents a single task collection period. P1 is an integer nonlinear optimization problem, which is usually difficult to solve directly. To solve P1 effectively, we analyze the system model and the optimization problem first. We find that the optimization problem can be decomposed into four sub-optimization problems, which are the relay selection decision problem, the channel allocation problem, the processing mode decision problem, and the task migration decision problem. The channel allocation problem and the processing mode decision problem are not related to other UAV states but are only related to the current UAV state. The UAVs can perform their own decisions. Thus, an algorithm based on multi-agent reinforcement learning can be used to solve it jointly.

3 QMIX-based solution algorithm

Since the states of all temporary UAVs at the next moment are only associated with the state of the current moment, they are independent of the states corresponding to the last moments. Therefore, the decision-making problem in this paper satisfies the Markovian properties. In the proposed multi-temporary UAV-assisted data collection network scenario, multiple temporary UAVs interact with the environment at the same time, and the uncertain behaviors among the UAVs will cause significant instability to the system. At the same time, limited by the means of information acquisition, it is difficult for the UAVs to obtain an accurate and real-time overall state of the environment in a large airspace. The decision-making center cannot control the temporary UAVs. Therefore, the relay selection and mission migration problem can be solved by the QMIX-based multi-agent deep reinforcement learning algorithm. In addition, the solution is described as a partially observable Markov decision process (POMDP) (Peng et al., 2020; Hossein Motlagh et al., 2016), which contains the following important parts.

3.1 Global states and locally observable states

To reduce the data transmission time and improve the real-time and accuracy of the data, the UAV can only obtain the immediate positional and directional information of other UAVs, while its trajectory information needs to be obtained by itself internally. s_t is used as the global state information.

$$s_t = \left(l_1, d_1, l_2, d_2, \ldots, l_j, d_j, \ldots, l_J, d_J, \delta\right).$$

Here, $l_j = (x_j, y_j, z_j)$ denotes the 3D position information of UAV j, d_j denotes the instantaneous heading angle information of UAV j, δ indicates whether the current task is a relay selection or a migration target selection, $\delta = 0$ indicates relay selection, and $\delta = 1$ indicates migration target selection. The accuracy of the environmental state information observed by the agents is negatively correlated with the distance between the agents, i.e., the greater the distance between the agents, the lower the accuracy of the acquired correlated states. The location information observed by the agent is a superimposition of the actual location information and environmental white noise.

Consequently, we designed a Gaussian noise distribution with a variance that changes with the actual distance between the agents, i.e., $N(0, \varpi D_{ab})$, where ϖ is a constant coefficient set at 0.02 and D_{ab} represents the line-of-sight distance between agents a and b. In detail, $D_{ab} \geq 50m$, where the observation information of agent a about agent b is superimposed with noise sampled from the Gaussian distribution $N(0, \varpi D_{ab})$. However, where $0m < D_{ab} < 50m$, the observational information of agent a regarding agent b does not contain the sampled noise. In summary, the locally observable information o_t^i of agent i mainly consists of the position coordinates l_t^o, heading angle d_t^o, its own position coordinates l_t^s, its own heading angle d_t^s, and its own number i superimposed on the noise in the other UAVs in the system. Assuming that there are N+1 UAVs in the system, the locally observable information o_t^i is as follows:

$$o_t^i = \left(l_t^{o,1}, d_t^{o,1}, l_t^{o,2}, d_t^{o,2}, \ldots, l_t^{o,N}, d_t^{o,N}, l_t^s, d_t^s, i, \delta\right).$$

3.2 Movement

Relay selection and migration target selection are both carried out for UAVs assuming that there are N UAVs in the system. The action of the agent i is denoted as $\mu_t^i = (\mu_t^{i,1}, \mu_t^{i,2}, \ldots, \mu_t^{i,N})$, $\mu_t^{i,n} \in \{0, 1\}$, and $\|\mu_t^i\|_1 \leq 1$. $\mu_t^{i,n} = 1$ chooses the UAV n. The joint action $\boldsymbol{\mu}_t = (\boldsymbol{\mu}_{t,1}, \boldsymbol{\mu}_{t,2}, \ldots, \boldsymbol{\mu}_{t,N})$ is the additive combination of the actions of all the agents, which is calculated as follows:

$$\boldsymbol{\mu}_t = \sum\nolimits_{i=1}^{N} \mu_t^i.$$

It can be expressed as follows:

$$\left(\boldsymbol{\mu}_{t,1}, \boldsymbol{\mu}_{t,2}, \ldots, \boldsymbol{\mu}_{t,N}\right) = \left(\sum\nolimits_{i=1}^{N} \mu_t^{i,1}, \sum\nolimits_{i=1}^{N} \mu_t^{i,2}, \ldots, \sum\nolimits_{i=1}^{N} \mu_t^{i,N}\right).$$

When a unique maximum element $\boldsymbol{\mu}_{t,n}^{max}$ exists for $\boldsymbol{\mu}_t$, performing a joint action $\boldsymbol{\mu}_t$ means selecting the UAV n. When there are a number of elements that are the same as the maximum element, the action $\boldsymbol{\mu}_t$ means that the UAV corresponding to one of the largest elements is randomly selected. For example, assuming that there are three UAVs numbered 1, 2, and 3 in the system, the movements of each UAV are shown as follows:

- $\mu_t^1 = (0, 0, 1)$
- $\mu_t^2 = (0, 1, 0)$
- $\mu_t^3 = (0, 0, 1)$

then, the joint action $\boldsymbol{\mu}_t = (0, 1, 2)$; executing $\boldsymbol{\mu}_t$ at this point means selecting UAV 3. Otherwise, if the movements of each UAV are

- $\mu_t^1 = (1, 0, 0)$
- $\mu_t^2 = (0, 1, 0)$
- $\mu_t^3 = (0, 0, 1)$

then, the joint action $\boldsymbol{\mu}_t = (1, 1, 1)$, and executing $\boldsymbol{\mu}_t$ at this point means selecting one UAV from those three UAVs randomly. Furthermore, if $\|\boldsymbol{\mu}_t\|_1 = 0$, the data are discarded.

3.3 Incentive

In multi-agent reinforcement learning, the design of the reward function significantly differs from that in single-agent reinforcement learning. We consider the design of the reward function from a global perspective, which is expressed as follows:

$$r_i = \eta\left(\frac{\lambda_1 S_i}{T_{tot,i}} + \frac{\lambda_2 S_i}{cost_i}\right).$$

In the formula, η denotes whether the task is successful, with $\eta = 1$ indicating success and otherwise indicating failure; λ_1, λ_2 denote weight coefficients; S_i denotes the original size of the data; $cost_i$ denotes the cost of task. It is important to note that the rewards here are for an individual task and not per time step t. In other words, agents do not receive immediate rewards after the joint action is performed. Instead, they must wait until the task is completed to obtain the rewards. The overall detailed flow of task processing via the QMIX-based selection strategy on the relay and migration target is shown in Figure 4.

As shown in Figure 4, QMIX is a novel value-based multi-agent reinforcement learning method. Each UAV makes the decisions with its own environment and sends the decisions to the mix network. The form of the QMIX fusion function is not directly assumed but is estimated through a neural network. Hence, the total action value function $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u})$ in QMIX is represented as follows:

$$Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) = MIX_{\hat{\theta}}\left(Q_1(\tau^1, u^1), \ldots, Q_n\left(\tau^n, u^n; \dot{\theta}\right)\right),$$

where $\hat{\theta}$ represents the parameters of the fusion function neural network. To meet the individual–global maximization criterion, QMIX requires the global action value function $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u})$ to be monotonic. It demands that the derivative of the global action value function $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u})$ with respect to each agent action value function $Q_i(\tau^l, u^l)$ be non-negative, which is mathematically described as follows:

$$\frac{\partial Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u})}{\partial Q_i(\tau^l, u^l)} = g_l \geq 0, \forall l \in N = \{1, 2, \ldots, n\},$$

where N represents the space of the agent indexes within the system. To ensure the monotonicity of $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u})$, QMIX incorporates parameters θ_h from the hypernetworks (Rashid et al., 2018), with the global state s as an input to generate the parameters for the fusion function neural network. The generated parameters are directly taken in absolute value to ensure the non-negativity of the fusion neural network parameters, thus ensuring the validity of the equation. The pseudocode for the QMIX-based solution algorithm is presented in Algorithm 1.

```
Initialize the experience pool, setting the upper and lower storage limits as $B_{max}$ and $B_{min}$ and the timestep counters as $C_1$ and $C_2$, setting $C_1 = C_2 = 0$.
Initialize the real neural network parameters for agents $\dot{\theta}$ and the fusion function neural network parameters $\hat{\theta}$; for the target neural network parameters for agents $\dot{\theta}^-$ and the fusion function neural network parameters $\hat{\theta}^-$, set $\dot{\theta}^- = \dot{\theta}$, $\hat{\theta}^- = \hat{\theta}$, and the hypernetwork parameters $\theta_h$.
For $e \in \{1,2,\ldots,E\}$ do:
    Initialize the state $\boldsymbol{s}_t = \boldsymbol{s}_0$, set $t = 0$.
    Initialize an empty list $L$ for historical sequences.
    While $s_t$ is not a terminal state, do:
        $C_1 = C_1 + 1$, $C_2 = C_2 + 1$
        For $l \in N = \{0,1,2,\ldots,n\}$ do:
            Under the global state $s_t$, agent $l$ observes its local observable state $o_t^l$.
            Based on its local state $o_t^l$, agent $l$ decides an action $\mu_t^l$ using an $\varepsilon-$greedy strategy.
        End for
        Combine the actions $\mu_t^l$ of all agents to form the joint action $\boldsymbol{\mu}_t = (\mu_t^1,\mu_t^2,\ldots,\mu_t^n)$.
        Execute the joint action $\boldsymbol{\mu}_t$, receive reward $r_t$, and complete the global state transition $s_t \rightarrow s_{t+1}$.
        Store the sequence $(s_t,\boldsymbol{o}_t,\boldsymbol{\mu}_t,r_t)$ in the historical sequence list $L$, where $\boldsymbol{o}_t = (o_t^1,o_t^2,\ldots,o_t^n)$.
        If the time step counter $C_1$ equals the preset training cycle, and the number of sequences in the experience pool is not less than $B_{min}$, then
            Randomly draw a batch of historical sequence segments from the experience pool.
            Each segment contains complete historical information from the initial state to the terminal state
            For $l \in N = \{0,1,2,\ldots,n\}$ do
                Combine $o_t^l$ and $\mu_t^l$ to form $\tau^l$ and $u^l$, and combine $o_{t+1}^l$ to form $\tau^{l\prime}$.
                Agent $l$ calculates the online neural network, based on $\tau^l$ and $u^l$ and its identifier $l$, and computes $Q_l(\tau^l,u^l;\dot{\theta})$.
                Agent $l$ calculates the target neural network, based on $\tau^{l\prime}$, and obtains $\max_{u^{l'}} Q_l(\tau^{l\prime},u^{l\prime};\dot{\theta}^-)$
            End for
            The hypernetwork takes the global state $s_t$ as input and output parameters, which are then taken in absolute value to obtain $\hat{\theta}$
            All $Q_l(\tau^l,u^l;\dot{\theta})$ are input to the online neural network of the fusion function to obtain $Q_{tot}(\boldsymbol{\tau},\boldsymbol{u};\theta)$
            All max $\max_{u^{l'}} Q_l(\tau^{l\prime},u^{l\prime};\dot{\theta}^-)$ are input to the target neural network of the fusion function to obtain $Q_{tot}^{max}(\boldsymbol{\tau}',\boldsymbol{u}';\theta^-)$.
            Compute the loss function $L(\theta)$ and optimize using gradient descent.
            Reset the time step counter $C_1 = 0$
        End if
        If the time step counter $C_2$ equals the update cycle for $\dot{\theta}^-$ and $\hat{\theta}^-$, then
            $\dot{\theta}^- = \dot{\theta}$, $\hat{\theta}^- = \hat{\theta}$.
            Reset the time step counter $C_2 = 0$
        End if
        Complete the global state transition, set $s_t = s_{t+1}$, $t = t + 1$.
    End while
    Store the historical sequence list $L$ in the experience pool.
End for
```

Algorithm 1. QMIX-based solution algorithm.

4 Simulation analysis

4.1 Parameter settings

In the simulation, we consider a power infrastructure data collection network environment assisted by multiple temporary UAVs. The maximum one-hop communication distance between the nodes is set to 30 m. In addition, the UAVs are configured with a communication radius of 300 m and a flight altitude of 50 m. Their starting locations and flight direction are given randomly. The CPU speed of the UAV is set to 1~3 GHz (GC = 10^9 cycles; GHz = GC/s), the raw data are set between 2 and 10 Mb, and the computational demand-to-data size ratio of the task ranges from 1 GC/Mb to 2 GC/Mb. The number of ground volunteer nodes is set to 30, which is denoted as $I = 30$. The starting location is given randomly, while the starting locations of the task nodes are known. The latency requirement of the task satisfies $T_{req,i} \sim U(1,5)s$ and $T_{req}^{th} = 100s$. For the QMIX-based multi-agent network, for each UAV, the hidden layers of the neural network are set as two layers 64 $*$ 64, the learning rate is 0.005, and the discount factor is 0.9. In terms of software, the simulation model was developed using Python 3.8 and TensorFlow 2.0. The experimental platform is a workstation based on the AMD R3600. More detailed parameters are shown in Table 3.

The simulation experiments are conducted from two aspects. On one hand, the convergence of the proposed algorithm is evaluated. On the other hand, the performances of the proposed algorithm and the comparative strategies under different operating conditions are observed by modifying variables such as the number of temporary UAVs N_u, the computational speed of the UAVs $\mathcal{C}_u$, and the transmission power of the UAVs p_{u0}.

4.2 Analysis of results

In this paper, the multi-agent deep reinforcement learning algorithm (QMIX) is used to find the optimal relay node selection and task migration target. The proposed algorithm is denoted as the QMIX-based selection strategy on relay and migration target (**QSSRMT**), and we design three groups of comparative schemes, which are as follows:

1) **ISSRMT** (Zhu et al., 2021): independent learning approach-based selection strategy on relay and migration target algorithm.

2) **DSSRMT** (Xia et al., 2021): deep Q-learning-based selection strategy on relay and migration target algorithm.
3) **VSSRMT** (Sunberg et al., 2016): value function decomposition method-based selection strategy on relay and migration target algorithm.

Figure 5 shows the reward variation curves for the number of UAVs $N_u = 5$, the computational speed of UAVs $C_u = 2GHz$, and the transmission power of UAVs $p_{u0} = 20dBm$. The blue curve represents the QSSRMT scheme, while the red curve represents the VDN-based VSSRMT scheme. Both schemes are multi-agent reinforcement learning algorithms based on the generic value function decomposition method, and the QSSRMT slightly outperforms the VSSRMT in terms of their reward value curves. The QMIX and VDN algorithms both fall under the category of value function decomposition methods. However, they differ significantly in their value function combination functions. QMIX utilizes a fusion neural network to approximate its fusion function, concurrently ensuring through parameter design that it upholds monotonicity, thereby complying with the IGM criterion. VDN simply believes that the fusion function is a summation function; i.e., the global action value function is equal to the summation of the action value functions of each agent. QMIX, which employs a neural network to approximate the fusion function, is more adaptable to the experimental environment of the model proposed in this chapter. In addition, the other orange and green curves in the figure indicate the DSSRMT and ISSRMT, respectively, and it can be found that both schemes perform relatively poorly. The DSSRMT employs the joint learning approach, treating the global environment as a whole and making centralized decisions for all agents. Essentially, the system transitions from a multi-agent framework to a single-agent one. The disadvantage of the joint learning method is that as the number of agents increases, the joint action value function will experience dimensional catastrophe, which ultimately leads to an unsolved problem. However, from the results of this experiment, the DSSRMT scheme can still provide a suboptimal solution to the optimization problem in this paper. The ISSRMT adopts the independent Q-learning method, where each agent learns its own action value function individually. According to the experimental results, it is difficult for ISSRMT to have a more stable performance in the experimental model.

Figure 6 shows the cost curves of various algorithms under the condition of UAV computational speed $C_u = 2GHz$ and UAV transmission power $p_{u0} = 20dBm$, with varying numbers of UAVs. Figure 5 shows that as the number of UAVs increases, the average cost of tasks decreases slightly. Having more UAVs can provide more computing resources, thereby reducing the impact of service request congestion. It also affords more opportunities and options for task migration, consequently lowering the task failure rate. A higher task success rate leads to a lower average cost per task.

Figure 7 illustrates the relationships between the system cost and different UAV transmission power levels. From the figure, we can find that as the UAV transmission power increases, there is a significant reduction in the average cost of task processing. The main reason is that the magnitude of the UAV transmission power influences the transmission rate of the communication link quality between the UAV and the ground nodes directly. As discerned from Eqs 1, 2, the larger the UAV transmission power, the higher the transmission rate of the downloading communication links. When the download rate is higher, the fine-grained task computation mode struggles to offer additional time-saving advantages; yet, it remains a more costly option. Under these conditions, adopting model 1 to simply package the data and then send them to the users directly becomes a more cost-effective choice.

Figure 8 shows the relationships between the average system cost and different UAV computational capacities. From the figure, we can find that when the UAV computational capacity is between 1 and 2 GHz, an increase in computational power leads to a rapid decrease in the average task cost. However, when the computational capacity exceeds 2 GHz, further increases have a negligible effect on the task processing cost. The enhancement in UAV computational capacity can reduce the time consumed for task processing, thereby decreasing the probability of task failure to a certain extent. However, once the task processing time is reduced to a certain level, it no longer remains the primary factor contributing to task failure. In other words, when the UAV has sufficient computational capacity, further reductions in task processing time no longer affect the task completion success rate significantly. Figure 9 shows the relationship between the average system cost and different numbers of ground nodes I; we set that the data size and latency requirement of the generation tasks of each node are the same. From the figure, we can find that when the number of ground volunteer nodes increases, the whole system cost increases; because more tasks need be processed, the system delays, and the cost increases. The performance of the proposed QMIX-based QSSRMT is evidently the best when compared with the other three algorithms.

5 Conclusion

In this paper, we investigate an efficient data collection and sharing strategy in outdoor data collection for power infrastructure networks aided by multiple temporary UAVs. A multi-constraint integer nonlinear optimization problem is established that jointly optimizes relay selection, channel allocation, task processing mode decision, and migration target selection. We propose a decision-making scheme via a QMIX-based multi-agent deep reinforcement learning method to accomplish the joint solutions of relay selection and migration target selection issues. Simulation experiments validate the effectiveness of the proposed strategies based on QMIX. For future work, we will investigate the cooperative data collection scheme by taking into account the UAVs and vehicles on the ground simultaneously for improving the efficiency.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material; further inquiries can be directed to the corresponding author.

Author contributions

QL: data curation, software, and writing–original draft. RX: software and writing–original draft. ZY: methodology, resources, and writing–review and editing. GW: formal analysis, investigation,

methodology, and writing–review and editing. ZH: data curation, software, and writing–review and editing. CY: supervision and writing–review and editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Conflict of interest

Authors QL, RX, and ZY were employed by Guangdong Power Grid Corporation. Authors GW and ZH were employed by Yunfu Power Supply Bureau, Guangdong Power Grid Corporation.

The remaining author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Hossein Motlagh, N., Taleb, T., and Arouk, O. (2016). Low-altitude unmanned aerial vehicles-based Internet of things services: comprehensive survey and future perspectives. *IEEE Internet Things J.* 3 (6), 899–922. doi:10.1109/jiot.2016.2612119

Hu, S., Zhao, X., Ren, Y., and Xu, D. (2021). "Decentralized formation tracking and disturbance suppression for collaborative UAVs transportation," in 2021 40th Chinese Control Conference (CCC), Shanghai, China, July, 2021, 5535–5539.

Hu, X., Wong, K. K., Yang, K., and Zheng, Z. UAV-assisted relaying and edge computing: scheduling and trajectory optimization. *IEEE Trans. Wirel. Commun.*, 2019, 18(10):4738–4752. doi:10.1109/twc.2019.2928539

Huang, W., Guo, H., and Liu, J. (2021). "Task offloading in UAV swarm-based edge computing: grouping and role division," in 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, December, 2021, 1–6.

Jayakody, D. N. K., Perera, T. D. P., Ghrayeb, A., and Hasna, M. O. (2020). Self-energized UAV-assisted scheme for cooperative wireless relay networks. *IEEE Trans. Veh. Technol.* 69 (1), 578–592. doi:10.1109/tvt.2019.2950041

Jeong, S., Simeone, O., and Kang, J. (2018). Mobile edge computing via a UAV-mounted cloudlet: optimization of bit allocation and path planning. *IEEE Trans. Veh. Technol.* 67 (3), 2049–2063. doi:10.1109/tvt.2017.2706308

Kumar, N., Vasilakos, A. V., and Rodrigues, J. J. P. C. (2017). A multi-tenant cloud-based DC nano Grid for self-sustained smart buildings in smart cities. *IEEE Commun. Mag.* 55 (3), 14–21. doi:10.1109/mcom.2017.1600228cm

L Abbate, C. (2023). A modern communications platform to enable the modern Grid: a utility-grade wireless broadband network. *IEEE Power Energy Mag.* 21 (5), 18–26. doi:10.1109/mpe.2023.3288589

Liu, B., Zhang, W., Chen, W., Huang, H., and Guo, S. (2020). Online computation offloading and traffic routing for UAV swarms in edge-cloud computing. *IEEE Trans. Veh. Technol.* 69 (8), 8777–8791. doi:10.1109/tvt.2020.2994541

Liu, Y., Zhou, J., Tian, D., Sheng, Z., Duan, X., Qu, G., et al. (2022). Joint communication and computation resource scheduling of a UAV-assisted mobile edge computing system for platooning vehicles. *IEEE Trans. Intelligent Transp. Syst.* 23 (7), 8435–8450. doi:10.1109/tits.2021.3082539

Lyu, F., Yang, P., Shi, W., Wu, H., Wu, W., Cheng, N., et al. (2019). "Online UAV scheduling towards throughput QoS guarantee for dynamic IoVs," in ICC 2019 - 2019 IEEE Internation-al Conference on Communications (ICC), Shanghai, China, May, 2019, 1–6.

Ng, J. S., Bryan Lim, W. Y., Dai, H. N., Xiong, Z., Huang, J., Niyato, D., et al. (2020). "Communication-efficient federated learning in UAV-enabled IoV: a joint auction-coalition approach," in GLOBECOM 2020 - 2020 IEEE Global Communications Conference, Taipei, Taiwan, December, 2020, 1–6.

Peng, H., Ye, Q., and Shen, X. (2020). Spectrum management for multi-access edge computing in autonomous vehicular networks. *IEEE Trans. Intelligent Transp. Syst.* 21 (7), 3001–3012. doi:10.1109/tits.2019.2922656

Pokhrel, S. R., Jin, J., and Vu, H. L. (2019). Mobility-aware multipath communication for unmanned aerial surveillance systems. *IEEE Trans. Veh. Technol.* 68 (6), 6088–6098. doi:10.1109/tvt.2019.2912851

Rahman, S., Akter, S., and Yoon, S. (2022). OADC: an obstacle-avoidance data collection scheme using multiple unmanned aerial vehicles. *Appl. Sci.* 12 (22), 11509. doi:10.3390/app122211509

Rashid, T., Samvelyan, M., de Witt, C. S., Farquhar, G., Foerster, J., and Whiteson, S.QMIX (2018). "Monotonic value function factorisation for deep multi-agent reinforcement learning," in 2018 35th International Conference on Machine Learning (ICML), Stockholm, SWEDEN, July, 2018.

Stoupis, J., Rodrigues, R., Razeghi-Jahromi, M., Melese, A., and Xavier, J. I. (2023). Hierarchical distribution Grid intelligence: using edge compute, communications, and IoT technologies. *IEEE Power Energy Mag.* 21 (5), 38–47. doi:10.1109/mpe.2023.3288596

Sun, L., Wan, L., Wang, J., Lin, L., and Gen, M. (2023a). Joint resource scheduling for UAV-enabled mobile edge computing system in Internet of vehicles. *IEEE Trans. Intelligent Transp. Syst.* 24, 15624–15632. Accepted. doi:10.1109/tits.2022.3224320

Sun, L., Wan, L., Wang, J., Lin, L., and Gen, M. (2023b). Joint resource scheduling for UAV-enabled mobile edge computing system in Internet of vehicles. *IEEE Trans. Intelligent Transp. Syst.* 24 (12), 15624–15632. doi:10.1109/tits.2022.3224320

Sunberg, Z. N., Kochenderfer, M. J., and Pavone, M. (2016). "Optimized and trusted collision avoidance for unmanned aerial vehicles using approximate dynamic programming," in 2016 IEEE International Conference on Robotics and Automation (ICRA), Stock-holm, Sweden, May, 2016, 1455–1461.

Wang, J., Zhang, X., He, X., and Sun, Y. (2023a). Bandwidth allocation and trajectory control in UAV-assisted IoV edge computing us-ing multiagent reinforcement learning. *IEEE Trans. Reliab.* 72, 599–608. Accepted. doi:10.1109/tr.2022.3192020

Wang, J., Zhang, X., He, X., and Sun, Y. (2023b). Bandwidth allocation and trajectory control in UAV-assisted IoV edge computing using multiagent reinforcement learning. *IEEE Trans. Reliab.* 72 (2), 599–608. doi:10.1109/tr.2022.3192020

Xia, Z., Du, J., Jiang, C., Wang, J., Ren, Y., and Li, G. (2021). "Multi-UAV cooperative target tracking based on swarm intelligence," in ICC 2021-IEEE International Conference on Communications, Montreal, QC, Canada, June, 2021, 1–6.

Zhang, S., Zhang, H., He, Q., Bian, K., and Song, L. Joint trajectory and power optimization for UAV relay networks. *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 161–164. 2018.doi:10.1109/lcomm.2017.2763135

Zhang, Z., Xie, X., Xu, C., and Wu, R. (2022). "Energy harvesting-based UAV-assisted vehicular edge computing: a deep reinforcement learning approach," in 2022 IEEE/CIC International Conference on Communications in China (ICCC Workshops), Sanshui, Foshan, China, August, 2022, 199–204.

Zhu, S., Gui, L., Zhao, D., Cheng, N., Zhang, Q., and Lang, X. (2021). Learning-based computation offloading approaches in UAVs-assisted edge computing. *IEEE Trans. Veh. Technol.* 70 (1), 928–944. doi:10.1109/tvt.2020.3048938