Joint Optimization Risk Factor and Energy Consumption in IoT Networks With TinyML-Enabled Internet of UAVs

Run Liu, *Graduate Student Member, IEEE*, Mande Xie, Anfeng Liu, and Houbing Song, *Fellow, IEEE*

***Abstract*—The high mobility of Internet unmanned aerial vehicles (IUAVs) has attracted attention in the field of data collection. With the rapid development of the Internet of Things (IoT), more and more data are generated by IoT networks. IUAV-aided IoT networks can efficiently collect data in specific areas, which is of great significance in disaster relief. In the data collection task, it is necessary to plan the flight trajectory for the data collector—IUAV, so that the IUAV can collect data efficiently. However, existing research basically only considers the efficiency of data collection by IUAVs, but rarely considers the safety of IUAVs during flight. Therefore, this article proposes an IUAV trajectory planning algorithm that integrates energy efficiency and safety using local search to address the issues mentioned above. At the same time, a tiny machine learning (TinyML) algorithm is designed to assist the IUAV in making real-time decisions during flight. First, we build a general mathematical model that describes the risk in a particular region. Then, consider guiding the IUAV to a safer trajectory by introducing virtual nodes in the flight trajectory. Furthermore, we designed a local search algorithm for the three tasks of IUAV access sequence, IoT networks cluster heads selection, and virtual nodes selection and solved them through iterative optimization. We also consider the unreachable situation of the virtual nodes and use TinyML technology to help the IUAV adjust the position of the virtual nodes in real time in case of an emergency. In the end, an IUAV trajectory is obtained that can efficiently collect IoT networks' data and fly safely. We have conducted a large number of simulation experiments to demonstrate the efficiency of the proposed algorithm compared to the baseline algorithm.**

***Index Terms*—Data collection, flight safety, Internet unmanned aerial vehicle (IUAV)-aided Internet of Things (IoT) networks, iterative optimization, local search, real-time tiny machine learning (TinyML) application.**

I. Introduction

INTERNET unmanned aerial vehicle (IUAV) has remarkable maneuverability and flexibility, enabling it to access areas that would be challenging for both humans and conventional vehicles to reach, and it is widely used in data collection, task offloading [1], [2], [3], crowd sensing [4], [5], health care [6], and service deployment [7], [8], [9], [10], [11]. In the IUAV-aided Internet of Things (IoT) network, these IUAVs can effectively help collect data [12], [13], making them an indispensable role in disaster relief [14].[1] When dispatching IUAVs to a designated area to collect data, both IUAVs and the IoT network must be fully prepared to maximize data collection efficiency. For IUAVs, it is necessary to plan efficient flight trajectories in advance so that they can collect data with high energy efficiency [15], [16], [17]. As for the IoT network, recommending a suitable cluster head node is crucial. The cluster head node should collect the data from the surrounding nodes in advance and offload the data after the IUAVs arrive near it. This approach can help ease the pressure on IUAVs during the data collection process [18], [19], [20].

However, when IUAVs are flying, it is more practical to consider planning the flight safety trajectory [21], [22]. Since IUAVs often fly in unsafe areas, meteorological and geographical conditions may pose threats[2] to the flight of IUAVs. This requires further consideration of the flight safety of IUAVs based on consideration of flight energy consumption and data collection efficiency.

Current research work on IUAV trajectory planning based on flight safety is often considered for specific risks and lacks a unified solution framework [21], [23]. Furthermore, existing research work on the use of IUAVs for data collection often ignores the consideration of IUAV flight safety [24]. Therefore, there is still less work on trajectory planning for energy consumption and flight risk in IUAV-aided IoT networks. According to the characteristics of the IUAV-aided IoT network itself, the trajectory of the IUAVs is closely related to the cluster head node in the IoT network, which brings difficulties to the trajectory planning of the IUAVs. At the same time, since the IUAV is an embedded device with limited computing power and storage space, the proposal of an efficient and lightweight algorithm under the condition of limited resources also presents great challenges in solving such problems.

Manuscript received 13 November 2023; accepted 22 December 2023. Date of publication 1 January 2024; date of current version 7 June 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62072475. *(Corresponding authors: Mande Xie; Anfeng Liu.)*

Run Liu and Anfeng Liu are with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: liurun22@mails.ucas.ac.cn; afengliu@csu.edu.cn).

Mande Xie is with the School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China (e-mail: xiemd@zjgsu.edu.cn).

Houbing Song is with the Department of Information Systems, University of Maryland at Baltimore County, Baltimore, MD 21250 USA (e-mail: songh@umbc.edu).

Digital Object Identifier 10.1109/JIOT.2023.3348837

[1] On 22 July 2021, the Wing Loong large-scale long-endurance IUAV equipped with a China Mobile wireless communication base station provided network security for Mihe Town, Gongyi City, Henan Province, China.

[2] Such as strong wind field environment, strong magnetic field environment, environment with many obstacles, no-fly zone, etc.

Based on the above considerations, this article proposes an IUAV trajectory design algorithm that jointly optimizes risk factors and energy consumption to solve the above problems. First, we established a unified risk model for the flight risk of IUAVs, so that IUAVs can deal with various risks that can be described by the model. Then, due to the coupling of IUAV trajectory planning and the selection of cluster head nodes of the IoT network, we decouple these two problems to reduce the difficulty of solving the problem. Next, we guide the IUAV to a safe trajectory by introducing virtual nodes on the ground. So, the IUAV planning problem is decomposed into three subproblems: 1) the selection of cluster heads; 2) the selection of virtual nodes; and 3) the access order of cluster head nodes and virtual nodes. For the three problems, we use local search technology to solve them separately, and then obtain the final optimal solution through iterative optimization. We model the algorithm as a Markov process and theoretically prove that the proposed algorithm obtains the global optimal solution with probability 1 when the Markov chain is long enough. At the same time, considering that during the IUAV flight process, the virtual node given in advance may be unreachable, and the virtual node needs to be adjusted in real time. Considering the application of machine learning in UAV [7], we proposed a tiny machine learning (TinyML) algorithm, so that the IUAV can give the most suitable virtual node location in real time according to the geographic location information.

Overall, the main contributions of this article are as follows.

1) Using the local search algorithm to iteratively optimize the cluster head node selection problem, the virtual node selection problem, and the node access order problem, the complexity of the algorithm is given. Compared to the exact algorithm, the algorithm significantly reduces the calculation time, while ensuring the superiority of the results.
2) A TinyML algorithm is proposed to enable IUAVs to pick a suitable virtual node location when the virtual node is unreachable. At the same time, the algorithm can ensure real-time performance while meeting the resource constraints of IUAVs.
3) Extensive simulations are conducted and the results show the high performance of the proposed algorithm.

The organization of the remainder of this article is as follows. Section II gives a review of related works. Section III introduces the system model and formulates the problem. Section IV proposed a local-search-based alternative optimal framework to solve the problem and a TinyML algorithm to enable IUAVs to pick a suitable virtual node in real time. Section V validates the effectiveness of the proposed algorithm through a large amount of simulations. Section VI gives the conclusion of this article.

II. Related Works

In this section, we present a brief review of existing works in three related areas: 1) cluster head selection in IUAV-aided IoT networks; 2) IUAV trajectory planning in risky environments; and 3) TinyML on embedded systems.

A. Cluster Head Selection in IUAV-Aided IoT Network

Cluster head election is a part of the clustering hierarchy protocol, which was first applied in the wireless sensor network [25], such as LEACH [26] and HEED [27]. Electing cluster heads can improve the efficiency of data collection. In IUAV-aided IoT networks, the cluster hierarchy protocol is applicable, and the election of cluster heads stands out as a crucial concern, with a plethora of research studies dedicated to addressing this matter.

In IUAV-aided IoT networks, energy consumption is the primary consideration, and cluster head election and IUAV trajectory optimization will be considered jointly. Zhu et al. [28] proposed a deep reinforcement learning with a sequential model strategy for solving IUAV trajectories and cluster head selection problems in IoT networks and the goal is to minimize the total energy consumption of the IUAV-IoT system. Ebrahimi et al. [29] proposed a data collection technique in IUAV-aided WSN using projection-based compressive data gathering (CDG) as a solution methodology. Ebrahimi et al. [29] emphasized achieving energy-efficient IUAV-assisted IoT data collection within IoT networks, where the collected data come with varying time constraints.

While many articles more or less address the reduction of energy consumption along with aspects like spectrum resources [30] and age of information [31], they have generally not taken into account the risks that IUAVs may encounter during their flight.

B. IUAV Trajectory Planning in Risky Environment

IUAVs will face many risks during flight, so when optimizing IUAV trajectory, we should not only consider the optimization goals related to system benefits but also consider the risks during IUAV flight.

In a stationary risk environment, Miller et al. [32] addressed optimal 3-D path planning for an IUAV by initially determining a risk optimal 2-D path for a fixed time problem, followed by solving a series of BVPs with varying IUAV speeds to find an admissible 2-D path that meets time and risk constraints. Finally, considering terrain relief, the authors approximate a 3-D path that minimizes the 2-D threat and meets additional constraints. Qu et al. [33] presented a method using partitioning, Dijkstra's algorithm, and potential field theory to address the problem of IUAV path planning in modern battlefield environments. In the context of IUAV path planning in a low-altitude dangerous environment with various obstacles and threats, Wen et al. [34] proposed a novel method, combining intuitionistic fuzzy set modeling for uncertainties, reachability set estimation for dynamic threats, subgoal selection for efficiency, and receding horizon planning for dynamic environments, resulting in improved online path-planning performance.

The IUAV trajectory planning problem in a risky environment can be difficult to solve, and most of the literature focuses only on solving this problem. Combining it with other problems will greatly increase the difficulty of solving, but the related research work is still very little.

TABLE I
SUMMARY OF KEY SYMBOLS

Symbol	Definition
$\mathcal{M}$	the set of clusters
M	the number of clusters
N	the number of IoT devices
$k(i)$	the number of IoT devices in the i-th cluster
$c_{m,n}$	binary variable indicating whether the n-th IoT device in the m-th cluster is elected as the cluster head
$P_{i,Los}$	the probability of LoS
$P_{i,NLos}$	the probability of NLoS
ϕ	the angle of attack
d_i	the Euclidean distance between IUAV and cluster head
H	the height when the IUAV is hovering
$\lambda_{LoS}, \lambda_{NLoS}$	the mean value of path loss of LoS link and NLoS link
B	communication bandwidth
$d_{m,n}$	the amount of data that the n-th IoT device in the m-th cluster needs to transmit
P_h	the power of hovering
m_{uav}	the mass of the IUAV
g	the acceleration due to gravity
r_p	the radius of the propeller
ρ	the air density
P_m	the horizontal flight power
L_{ij}	binary variable indicating whether the IUAV flies from the i-th cluster to the j-th cluster.
P_c	the communication power
$v_{i,j}^k$	a virtual node that belongs to the k-th cluster head
$\phi_c(c_{i,j})$	the operator of the LS-CHS
$\phi_v(v_{i,j}^k)$	the operator of the LS-VS
$\phi_t(\boldsymbol{o}_i)$	the operator of the LS-TP

C. TinyML on Embedded Systems

TinyML empowers embedded devices, and how to ensure high performance is particularly important when resources are limited. Giordano et al. [35] introduced an energy-efficient proof-of-concept device that utilizes Bluetooth low energy (BLE) and NFC connectivity to continuously monitor and detect various construction tasks and potential misuse of handheld power tools. The device incorporates a TinyML algorithm for onboard data processing, achieving high efficiency and low latency. Pavan et al. [36] presented a TinyML solution that uses ultra-wideband radar for the detection of presence with privacy, introducing specialized neural networks to accommodate the memory and processing limitations of small devices, with successful testing in indoor and in-car scenarios. It can be seen that TinyML has a bright future in embedded devices. As an embedded device, the combination of the Internet of IUAV and TinyML can obtain greater benefits.

III. SYSTEM MODEL AND PROBLEM FORMULATION

This section introduces the framework of the system to be studied as well as the required mathematical models, and finally formalizes the research problem. The key symbols are listed in Table I.

A. System Model

1) Network Model: The network considered in this article is IUAV-aided IoT networks. The IoT nodes in the network are clustered according to their geographical location. Each cluster elects a cluster head according to a protocol, and the rest of the nodes in the cluster send information to the cluster head. Suppose that there are N IoT devices in the network, they are divided into M clusters according to their geographic location, $\mathcal{M} = \{1, \ldots, M\}$. Use $k(i)$ to indicate the number of devices in the ith cluster, and use $c_{m,n} = 1$ to indicate that the nth IoT device in the mth cluster is elected as the cluster head; otherwise, $c_{m,n} = 0$, so we have the following constraints:

$$\sum_{j=1}^{k(i)} c_{i,j} = 1 \quad \forall i \in \mathcal{M} \tag{1}$$

$$\sum_{i=1}^{M} k(i) = N. \tag{2}$$

2) Communication Model: In this article, we use the air-to-ground (A2G) model as the communication model between IUAVs and IoT devices [37]. For the A2G communication model, when the IUAV communicates with the cluster head, the probability of Line of Sight (LoS) and the probability of Non-LoS (NLoS) are, respectively, recorded as $P_{i,\mathrm{LoS}}$ and $P_{i,\mathrm{NLoS}}$, the calculation method of $P_{i,\mathrm{LoS}}$ is

$$P_{i,\mathrm{LoS}} = \frac{1}{1 + a\exp[-b(\phi - a)]} \tag{3}$$

where a and b are determined by the environment of the drone, $\phi = 180/\pi \times \sin^{-1}(H/d_i)$ represents the angle of attack, d_i represents the Euclidean distance between IUAV and cluster head, and H represents the height when the IUAV is hovering. Therefore, the probability of NLoS is $P_{i,\mathrm{NLoS}} = 1 - P_{i,\mathrm{LoS}}$. So, the average path loss between cluster head and IUAV is

$$\bar{P}_{i,\mathrm{Loss}} = P_{i,\mathrm{LoS}}(K_0 + \lambda_{\mathrm{LoS}}) + P_{i,\mathrm{NLoS}}(K_0 + \lambda_{\mathrm{NLoS}}) \tag{4}$$

where λ_{LoS} and λ_{NLoS} represent the mean value of path loss of LoS link and NLoS link, respectively, $K_0 = 10k\log_{10}(4\pi f_c H/c)$, k represents the path-loss index, c represents the speed of light, and f_c represents the carrier frequency. So the average data transmission rate r_i^t between IUAV and cluster head can be calculated by the Shannon formula

$$r_i^t = B\log_2\left(1 + \frac{P}{\bar{P}_{i,\mathrm{Loss}} N_0}\right) \tag{5}$$

where B is the communication bandwidth, P is the transmission power, and N_0 is the noise power spectral density.

3) Energy Model: Due to the focus of this article on data collection in IUAV-aided IoT networks, in each round of data collection, other nodes within the cluster only need to send their data once to the cluster head. The cluster head receives the data sent from other nodes. Let $d_{m,n}$ represent the amount of data that the nth IoT device in the mth cluster needs to transmit. We assume that the energy consumption of IoT devices follows the wireless hardware energy dissipation model, where the transmitter dissipates energy to operate the wireless electronic devices and power amplifier, and the receiver dissipates energy to operate the wireless electronic devices. The free-space and multipath fading channel models [38] are used on the basis of the distance between the transmitter and receiver. Specifically, the transmission of $d_{m,n}$ bits of data requires energy

$$E_{m,n}^{tx} = \begin{cases} d_{m,n}E_e + d_{m,n}\epsilon_{\mathrm{fs}}d^2, \ d < d_o \\ d_{m,n}E_e + d_{m,n}\epsilon_{\mathrm{mp}}d^4 \ d \geq d_o. \end{cases} \tag{6}$$

The reception of $d_{m,n}$ bits of data requires energy

$$E_{m,n}^{rx} = d_{m,n} E_{\mathrm{e}} \tag{7}$$

here, E_{e} depends on factors, such as signal digital encoding, modulation, filtering, and spreading, while the amplifier energy e_{fs} or e_{mp} depends on the distance to the receiver and the acceptable bit error rate. For this article, communication energy parameters are set as follows: $E_{\mathrm{e}} = 50$ nJ/bit, $\epsilon_{\mathrm{fs}} = 10$ pJ/bit/m^2, and $\epsilon_{\mathrm{mp}} = 0.0013$ pJ/bit/m^4 [39].

The energy consumption of an IUAV can be divided into three parts: 1) propulsion energy during movement; 2) hovering energy during communication with ground devices; and 3) energy consumed during communication with ground devices. Whether in flight or hovering, the IUAV's rotor needs to overcome gravity to maintain a certain height, and the power required for this, denoted as P_h, is given by the following formula [29]:

$$P_h = \sqrt{\frac{(m_{\mathrm{uav}} g)^3}{2\pi r_p^2 n_p \rho}} \tag{8}$$

where m_{uav} is the mass of the IUAV, g is the acceleration due to gravity, r_p is the radius of the propeller, n_p is the number of propellers, and ρ is the air density.

The horizontal flight power P_m of the IUAV is directly proportional to its moving speed v_{uav} and can be expressed as

$$P_m = \frac{P_{\max} - P_{\mathrm{idle}}}{v_{\max}} v_{\mathrm{uav}} + P_{\mathrm{idle}} \tag{9}$$

where $v_{\max}$ represents the maximum speed of the IUAV, $P_{\max}$ represents the power consumption at full speed, and P_{idle} represents the power consumption when the IUAV is in an idle state.

Considering that the IUAV flies at a height H, and the coordinate of the cluster head in the mth cluster is given by $(x_i, y_i, 0)$, and the coordinate of the IUAV during data collection is (x_j, y_j, H), the Euclidean distance d_{ij} between the IUAV flying from the ith cluster to the jth cluster is calculated as

$$d_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}. \tag{10}$$

Let T be the total flight time of the IUAV, which is calculated as

$$T = \frac{1}{v_{\mathrm{uav}}} \sum_{i=0}^{c+1} \sum_{j=0, j \neq i}^{c+1} d_{ij} x_{ij} \tag{11}$$

where x_{ij} represents whether the IUAV flies from the ith cluster to the jth cluster. If $x_{ij} = 1$, it means the IUAV flies from the ith cluster to the jth cluster; if $x_{ij} = 0$, it means there is no adjacent visit order between the ith and jth clusters. Then, the total flight energy E_f of the IUAV is given by

$$E_f = T(P_h + P_m). \tag{12}$$

This equation shows that the flight energy of the IUAV includes the energy consumed to maintain the height H of the rotor and the energy used for propulsion during movement.

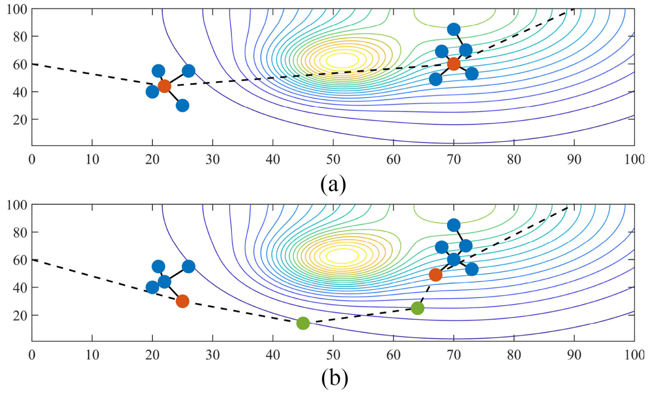


Fig. 1. Function of the virtual nodes. (a) Original. (b) Adjusted.

The energy consumption E_i for collecting data from the cluster head in the ith cluster is given by the formula

$$E_i = \frac{\sum_{j=1}^{d(i)} d_{i,j}}{r_i^t} (P_h + P_c) \tag{13}$$

where P_c represents the power consumption when the IUAV communicates with IoT devices, which depends on hardware equipment. Therefore, the total energy consumed by the IUAV to collect data from all cluster heads in the network is

$$E_u = \sum_{i=1}^{c} E_i + E_f. \tag{14}$$

4) Risk Model and Virtual Node: The terrain has a side length of L, and it is divided into grids with a step size of τ. Thus, $\lceil L/\tau \rceil \times \lceil L/\tau \rceil$ grids can be formed. If the IUAV's flight path is denoted as ι, and during its flight, it passes through some of the grids, we can represent the set of grids it passes through as $\Omega(\iota)$. Then, the cumulative risk level of the flight path corresponding to this set of grids is defined as

$$R[\Omega(\iota)] = \sum_{i \in \{1, \ldots, |\Omega(\iota)|\}} r_i \tag{15}$$

where r_i represents the risk value of the ith grid that the IUAV's trajectory ι passes through. Let $\mathcal{V} = \{1, \ldots, \lceil L/\tau \rceil\}$, and use $v_{i,j}^k$ to represent a virtual node that belongs to the kth cluster head, where $k \in \mathcal{M}$ and $i, j \in \mathcal{V}$.

The function of the virtual node is to guide the IUAV closer to a safer trajectory, as shown in Fig. 1. In Fig. 1(a), the IUAV will encounter a high-risk area when flying from one cluster head to another. From (15), it can be calculated that the risk level of the current flight path is relatively high. In Fig. 1(b), the green nodes are virtual nodes. Flying the IUAV along the virtual nodes will reduce the risk of the flight path of the IUAV. Therefore, how to plan the location of virtual nodes is also an issue that needs to be considered.

B. Problem Formulation

The objective of this article is to plan the path of IUAV in a way that minimizes both the risk that IUAV faces and the energy consumption of the IoT network while collecting data. Since our optimization targets include the IUAV's energy consumption, the energy consumption of the IoT network, and the IUAV's flight safety factor, it becomes a multiobjective optimization problem. To simplify the multiobjective optimization problem, we use a method of normalizing, weighting, and summing the objective functions to transform it into a single-objective optimization problem, formalized as

$$\mathbf{P1}: \min_{\boldsymbol{x},\boldsymbol{u},\boldsymbol{c},\boldsymbol{v}} \quad \alpha E_n + \beta E_u + \gamma R \tag{16}$$

$$\text{s.t.} \quad \sum_{j=1}^{k(i)} c_{i,j} = 1 \quad \forall i \in \mathcal{M} \tag{16a}$$

$$c_{i,j} \in \{0,1\} \quad \forall i,j \in \mathcal{M} \tag{16b}$$

$$\sum_{i\in\mathcal{H}} x_{ij} = 1 \quad \forall j \in \mathcal{H} \tag{16c}$$

$$\sum_{j\in\mathcal{H}} x_{ij} = 1 \quad \forall i \in \mathcal{H} \tag{16d}$$

$$u_j > u_i + n\big(x_{ij} - 1\big) + 1 \quad \forall i,j \in \mathcal{H}\backslash\{1\} \tag{16e}$$

$$x_{ij} \in \{0,1\} \quad \forall i,j \in \mathcal{H} \tag{16f}$$

$$u_i > 0 \quad \forall i \in \mathcal{H} \tag{16g}$$

$$\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}} v_{i,j}^k = K \quad \forall k \in \mathcal{M} \tag{16h}$$

$$v_{i,j}^k \in \{0,1\} \quad \forall i,j \in \mathcal{V}\forall k \in \mathcal{M}. \tag{16i}$$

In optimization problem **P1**, the objective function incorporates three components: 1) the IUAV energy consumption; 2) the energy consumption of the IoT network; and 3) the IUAV flight safety factor. Each of these components is multiplied by corresponding weights, which are manually chosen based on their significance in different scenarios.

Constraints (16a) and (16b) ensure that each cluster can have only one cluster head, and the decision variables for these constraints are binary (0-1 variables). Constraints (16c)–(16g) impose that the IUAV can visit each cluster exactly once, and there must be no subcycles in its path. These constraints are formulated using the Miller–Tucker–Zemlin (MTZ) approach [40].

Constraints (16h) and (16i) stipulate that, excluding the last cluster, the total number of virtual nodes belonging to all clusters should be K, and the decision variables for these constraints are also binary (0-1 variables).

Since the objective function is nonlinear and the constraints involve both integer and noninteger variables, problem **P1** is categorized as a mixed-integer nonlinear programming problem (MINLP). It is known to be NP-Hard, which implies that no polynomial-time algorithm can solve it.

In the next section, we will devise an approximate algorithm based on local search to find an approximate solution for problem **P1**, considering its specific formulation.

Proposition 1: Based on the model and the problem formulation, the solution space of the problem to be studied is

$$\prod_{i=1}^{M} |\mathcal{V}|^{2|\mathcal{V}_i|} \prod_{i=1}^{M} k(i) \left(M + \sum_{i=1}^{M} |\mathcal{V}_i|\right)!.$$

Proof: First, the number of virtual nodes owned by the cluster i is $|\mathcal{V}_i|$, and there are $|\mathcal{V}|^2$ positions that each virtual node can choose. So for the stage of virtual node location selection, the size of the solution space with M clusters is $\prod_{i=1}^{M} |\mathcal{V}|^{2|\mathcal{V}_i|}$. Then, the IoT devices for cluster i are $k(i)$, and each cluster can only have one cluster head. Therefore, for the stage of selection of the cluster head, the size of the solution space with M clusters is $\prod_{i=1}^{M} k(i)$. Next, the IUAV must visit all cluster heads and virtual nodes. The number of cluster heads and virtual nodes in the network is $M + \sum_{i=1}^{M} |\mathcal{V}_i|$, and the number of combinations of the full arrangement of these $M + \sum_{i=1}^{M} |\mathcal{V}_i|$ nodes is $(M + \sum_{i=1}^{M} |\mathcal{V}_i|)!$, i.e., for the stage of IUAV path planning, the size of the solution space is $(M+\sum_{i=1}^{M} |\mathcal{V}_i|)!$. The solution space of the above three stages is multiplied and the following solution space of the problem studied in this article is obtained:

$$\prod_{i=1}^{M} |\mathcal{V}|^{2|\mathcal{V}_i|} \prod_{i=1}^{M} k(i) \left(M + \sum_{i=1}^{M} |\mathcal{V}_i|\right)!.$$

■

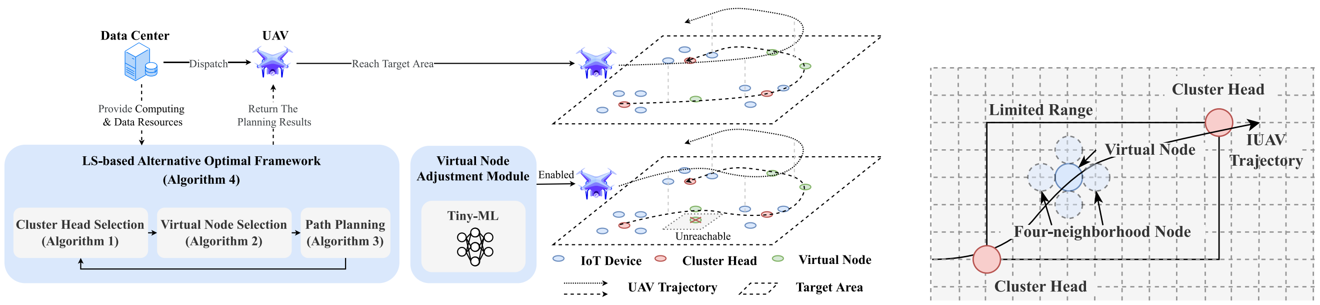
From Proposition 1, we theoretically analyze the solution space of the problem. Due to the huge solution space of the problem studied in this article, it is a difficult problem to solve. Since the subproblem of the problem studied in this article—IUAV trajectory planning problem—can be simplified to a traveling salesman problem which is a well-known NP-hard problem, the problem to be studied is also an NP-hard problem.

IV. Algorithm Design

In this section, we present the detailed design of the proposed algorithm. The overall framework of the algorithm is shown in Fig. 2(a). As can be seen from the previous section, problem **P1** is an MINLP, and there is no polynomial-time algorithm that can solve it quickly. The characteristic of this problem is that it can be decomposed into three subproblems cluster head selection $\{\boldsymbol{c}\}$, virtual node selection $\{\boldsymbol{v}\}$, and IUAV trajectory $\{\boldsymbol{x}, \boldsymbol{u}\}$. These problems have the following characteristics.

1) *The solution space is large.* The solution space is multiplicative (Proposition 1).
2) *Known conditions are not helpful for problem solving.* The conditions of each risk map are unique and only useful for solving the UAV flight trajectory for that particular map. However, this does not provide a general solution for UAV trajectories, as different risk maps will have different conditions that must be taken into account.
3) *There are requirements on the time complexity of the algorithm.* Accurate results need to be achieved in algorithms of exponential magnitude, but the solution time is unacceptable.
4) *Approximate solutions are acceptable.* The nature of the problem to be solved is highly nonconvex, and it is difficult to obtain accurate results. Due to this, we try to find an approximately optimal solution.

Due to the above characteristics of the problem, we decided to use a local search method to solve the problem. In this article, it is decomposed into three subproblems by applying


Data Center
UAV
Dispatch
Reach Target Area
Provide Computing & Data Resources
Return The Planning Results
LS-based Alternative Optimal Framework (Algorithm 4)
Cluster Head Selection (Algorithm 1)
Virtual Node Selection (Algorithm 2)
Path Planning (Algorithm 3)
Virtual Node Adjustment Module
Tiny-ML
Enabled
Unreachable
IoT Device
Cluster Head
Virtual Node
UAV Trajectory
Target Area
Cluster Head
Limited Range
Virtual Node
IUAV Trajectory
Four-neighborhood Node
Cluster Head

(a) (b)

Fig. 2. (a) Algorithm framework. (b) Illustration of virtual node selection.

the iterative optimization method, and the local search algorithm is used to alternately optimize the cluster head selection $\{\boldsymbol{c}\}$, virtual node selection $\{\boldsymbol{v}\}$, and IUAV trajectory $\{\boldsymbol{x}, \boldsymbol{u}\}$. Finally, an alternative optimal local search algorithm for the energy consumption and flight risk minimization problem is summarized.

A. Stage I: Cluster Head Selection

For given $\{\boldsymbol{v}, \boldsymbol{x}, \boldsymbol{u}\}$, the cluster head selection can be optimized by solving **P2**

$$\mathbf{P2:} \min_{\boldsymbol{c}} \quad \alpha E_n + \beta E_u + \gamma R \tag{17}$$

$$\text{s.t.} \quad \sum_{j=1}^{k(i)} c_{i,j} = 1 \quad \forall i \in \mathcal{M} \tag{17a}$$

$$c_{i,j} \in \{0, 1\} \quad \forall i, j \in \mathcal{M}. \tag{17b}$$

Due to the nonlinearity of the objective function and the integer constraints of (17b), this is an integer nonlinear program. For this problem, we designed a Local Search algorithm to quickly find a high-quality approximate solution. We design the operator using the Delaunay triangulation algorithm. First, do a Delaunay partition according to the positions of all nodes in the cluster. Then, other nodes connected to the node are the neighbors of the node. When an operator operates on a node, it will randomly select a node from the neighbors of the node. The reason why Delaunay triangulation is used to construct the neighbors of nodes is because Delaunay triangulation has the property that it forms a triangle with the nearest three points and each line segment (edge of the triangle) does not intersect. This ensures that when the operator operates on a node, the operation result is related to the adjacent nodes of the current node. Use $\mathcal{D}_{i,j}^c$ to represent the set of adjacent nodes calculated by the Delaunay triangulation algorithm for the jth node of the ith cluster. Then define the operator

$$\phi_c\left(c_{i,j}\right) \in \mathcal{D}_{i,j}^c. \tag{18}$$

This operator randomly selects an element from the set $\mathcal{D}_{i,j}^c$ as an output, and the probability that any element in the set $\mathcal{D}_{i,j}^c$ is selected is $(1/|D_{t,j}^c|)$. When using the Local Search algorithm, first randomly initialize a solution $\boldsymbol{c}_{ini}$, and then use the operator $\phi_c(\cdot)$ to perturb the solution $\boldsymbol{c}$ when the termination condition is not reached, and observe the results. If the new result is better, it is accepted; otherwise, the Metropolis acceptance criterion [41] is accepted with probability (Algorithm 1). The specific algorithm pseudocode is shown in Algorithm 2.

```
Algorithm 1: MAC($s_{new}, s_{ini}, T$)
Input: $s_{new}, s_{ini}, T$
Output: $s_{opt}$
if $s_{new} < s_{ini}$ then
    $s_{opt} \leftarrow s_{new}$.
else
    $p \leftarrow U(0, 1)$.
    if $p < \exp\left[\frac{(s_{ini} - s_{new})}{s_{new} \cdot T}\right]$ then
        $s_{opt} \leftarrow s_{new}$.
    end
end
return $s_{opt}$.
```

```
Algorithm 2: LS-Based Cluster Head Selection Algorithm
Input: Virtual node $\{\boldsymbol{v}\}$, IUAV trajectory $\{\boldsymbol{x}, \boldsymbol{u}\}$, initial control
       parameter $\delta_s^c$, stop control parameter $\delta_e^c$, attenuation
       factor $\alpha^c$.
Output: Cluster head $\{\boldsymbol{c}\}$.
1  $T \leftarrow \delta_s^c$.
2  Initialize the cluster head selection decision $\boldsymbol{c}_{ini}$.
3  Use Delaunay triangulation algorithm to construct the neighbors
   of nodes in each cluster.
4  while $T > \delta_e^c$ do
5      while the number of iteration is less than $L_c$ do
6          for each cluster do
7              Perturb the original solution $c_{i,j}^{ini} \in \boldsymbol{c}_{ini}$ with
               operator $\phi_c(c_{i,j}^{ini})$ to get a new solution $c_{i,j}^{new} \in \boldsymbol{c}_{new}$.
8          Calculate the objective function value according to
           $\{\boldsymbol{c}_{ini}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{u}\}$ and $\{\boldsymbol{c}_{new}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{u}\}$ respectively, and the
           values are denoted as $a_{ini}^c$ and $a_{new}^c$.
9          $c_{opt} \leftarrow MAC(\boldsymbol{c}_{new}, \boldsymbol{c}_{ini}, T)$.
10     $T = T \times \alpha^c$.
11 return $\boldsymbol{c}_{opt}$.
```

The computing complexity of Algorithm 2 is analyzed as follows. First, in the worst case, the number of iterations of the outer loop is $L_c \lceil \log_\alpha (\delta_e/\delta_s) \rceil$. The time complexity of lines 6 and 7 is $\mathcal{O}(M)$ and the time complexity of line 8 is

$\mathcal{O}(C)$. So, the time complexity of Algorithm 2 is $\mathcal{O}[(M+C)L_c\lceil\log_{\alpha}(\delta_e/\delta_s)\rceil]$.

B. Stage II: Virtual Node Selection

For given $\{\boldsymbol{c},\boldsymbol{x},\boldsymbol{u}\}$, the virtual node selection problem can be optimized by solving **P3**

$$\textbf{P3}:\min_{\boldsymbol{v}}\quad \alpha E_n+\beta E_u+\gamma R \tag{19}$$

$$\text{s.t.}\quad \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}} v_{i,j}^{k}=K\quad \forall k\in\mathcal{M} \tag{19a}$$

$$v_{i,j}^{k}\in\{0,1\}\quad \forall i,j\in\mathcal{V}\quad \forall k\in\mathcal{M}. \tag{19b}$$

Similarly, **P3** is also integer nonlinear programming. When designing the local search algorithm of **P3**, the four neighbors of any position of the $\lceil L/\tau\rceil\times\lceil L/\tau\rceil$ grids on the risk map are taken as its neighbors. Given the limited range of the IUAV, if only the four-neighborhood of a certain grid position is taken into account, the feasible set of the problem is infinite, which does not meet the requirement of a limited target range. To address this, we introduce the following constraints. The range of the selected virtual nodes must be between the cluster head nodes p and k in the adjacent access order, and the virtual node must belong to the cluster head k. The decision variable $v_{i,j}^{k}$ indicates that when selecting a node, the position of the node must not exceed the rectangular frame formed by the cluster head nodes in the adjacent access sequence, as shown in Fig. 2(b). Then define the operator

$$\phi_v\left(v_{i,j}^{k}\right)\in\mathcal{D}_k^{v}. \tag{20}$$

This operator randomly selects an element from the set $\mathcal{D}_k^{v}$ as an output, and the probability that any element in the set $\mathcal{D}_k^{v}$ is selected is $(1/|\mathcal{D}_k^{v}|)$. When using the Local Search algorithm, first randomly initialize a solution $\boldsymbol{v}_{ini}$, and then use the operator $\phi_v(\cdot)$ to perturb the solution $\boldsymbol{v}$ when the termination condition is not reached, and observe the results. If the new result is better, it is accepted; otherwise, the Metropolis acceptance criterion [41] is accepted with probability. The specific algorithm pseudocode is shown in Algorithm 3.

The computing complexity of Algorithm 3 is $\mathcal{O}[(\sum_{i=1}^{M}|\mathcal{V}_i|+C)L_v\lceil\log_{\alpha}(\delta_e/\delta_s)\rceil]$.

C. Stage III: IUAV Path Planning

For given $\{\boldsymbol{c},\boldsymbol{v}\}$, the IUAV path-planning problem can be optimized by solving **P4**

$$\textbf{P4}:\min_{\boldsymbol{x},\boldsymbol{u}}\quad \alpha E_n+\beta E_u+\gamma R \tag{21}$$

$$\text{s.t.}\quad \sum_{i\in\mathcal{H}} x_{ij}=1\quad \forall j\in\mathcal{H} \tag{21a}$$

$$\sum_{j\in\mathcal{H}} x_{ij}=1\quad \forall i\in\mathcal{H} \tag{21b}$$

$$u_j>u_i+n\left(x_{ij}-1\right)+1\quad \forall i,j\in\mathcal{H}\backslash\{1\} \tag{21c}$$

$$x_{ij}\in\{0,1\}\quad \forall i,j\in\mathcal{H} \tag{21d}$$

$$u_i>0\quad \forall i\in\mathcal{H}. \tag{21e}$$

Algorithm 3: LS-Based Virtual Node Selection Algorithm

```
Input: Cluster head $\{\boldsymbol{c}\}$, IUAV trajectory $\{\boldsymbol{x},\boldsymbol{u}\}$, initial control
       parameter $\delta_s^{v}$, stop control parameter $\delta_e^{v}$, attenuation
       factor $\alpha^{v}$.
Output: Virtual node $\{\boldsymbol{v}\}$.
$T\leftarrow\delta_s^{v}$.
According to the position of the cluster head $\{\boldsymbol{c}\}$, plan the
feasible area of the virtual node.
Initialize the virtual node selection decision $\boldsymbol{v}_{ini}$.
while $T>\delta_e^{v}$ do
    while the number of iteration is less than $L_v$ do
        for each cluster do
            Perturb the original solution $v_{i,j}^{k,ini}\in\boldsymbol{v}_{ini}$ with
            operator $\phi_v(v_{i,j}^{k,ini})$ to get a new solution
            $c_{i,j}^{k,new}\in\boldsymbol{v}_{new}$.
        Calculate the objective function value according to
        $\{\boldsymbol{c},\boldsymbol{v}_{ini},\boldsymbol{x},\boldsymbol{u}\}$ and $\{\boldsymbol{c},\boldsymbol{v}_{new},\boldsymbol{x},\boldsymbol{u}\}$ respectively, and the
        values are denoted as $a_{ini}^{c}$ and $a_{new}^{c}$.
        $\boldsymbol{v}_{opt}\leftarrow MAC(\boldsymbol{v}_{new},\boldsymbol{v}_{ini},T)$.
    $T=T\times\alpha^{v}$.
return $\boldsymbol{v}_{opt}$.
```

Since this problem is similar to the traveling salesman problem, we set the visit sequence of cluster heads and virtual nodes to $\boldsymbol{o}_i$ in this problem. For such access sequences, an effective perturbation method is the 2-opt method. That is, randomly select a route, then randomly select two nodes that are not connected on the route $\boldsymbol{o}_i$, flip the path between the two nodes to obtain a new path $\boldsymbol{o}_{i+1}$, and then map $\boldsymbol{o}_{i+1}$ to $\{\boldsymbol{x},\boldsymbol{u}\}$. Then define the operator

$$\phi_t(\boldsymbol{o}_i)\in\mathcal{D}_i^{t}. \tag{22}$$

This operator randomly selects an element from the set $\mathcal{D}_i^{t}$ as an output, and the probability that any element in the set $\mathcal{D}_i^{t}$ is selected is $1/|\mathcal{D}_i^{t}|$. When using the Local Search algorithm, first randomly initialize a solution v, and then use the operator $\phi_t(\cdot)$ to perform 2-opt perturbation on the solution $\boldsymbol{o}_i$ when the termination condition is not reached, and observe the results. If the new result is better, it is accepted; otherwise, the Metropolis acceptance criterion [41] is accepted with probability. The specific algorithm pseudocode is shown in Algorithm 4.

Similar as Algorithm 3, the computing complexity of Algorithm 4 is $\mathcal{O}[CL_t\lceil\log_{\alpha}(\delta_e/\delta_s)\rceil]$.

D. Alternative Optimal Algorithm

The iterative optimization algorithm iterates the optimization Algorithms 3 and 4 in turn and updates the current optimal solution during the iterative process to obtain the final optimal solution. Algorithm pseudocode see Algorithm 5. If $L_c=L_v=L_c=L$, the computing complexity of Algorithm 5 is $\mathcal{O}[(M+\sum_{i=1}^{M}|\mathcal{V}_i|+3C)n_tL\lceil\log_{\alpha}(\delta_e/\delta_s)\rceil]$.

E. Analysis of LS-Based Algorithm

Theorem 1: The LS-CHS, LS-VS, and LS-TP algorithms correspond to a Markov chain, respectively.

Algorithm 4: LS-Based Trajectory Planning Algorithm

```
Input: Cluster head $\{\boldsymbol{c}\}$, Virtual node $\{\boldsymbol{v}\}$, initial control
       parameter $\delta_s^p$, stop control parameter $\delta_e^p$, attenuation
       factor $\alpha^p$.
Output: IUAV trajectory $\{\boldsymbol{x}, \boldsymbol{u}\}$.
$T \leftarrow \delta_s^p$.
Initialize the flight order $\boldsymbol{o}_{ini}$.
while $T > \delta_e^p$ do
    while the number of iteration is less than $L_p$ do
        Perturb the original solution $\boldsymbol{o}_{ini}$ with operator $\phi_t(\boldsymbol{o}_{ini})$
        to get a new solution $\boldsymbol{o}_{new}$.
        Convert $\boldsymbol{o}_{ini}$ and $\boldsymbol{o}_{new}$ to $\boldsymbol{x}_{ini}, \boldsymbol{u}_{ini}$ and $\boldsymbol{x}_{new}, \boldsymbol{u}_{new}$.
        Calculate the objective function value according to
        $\{\boldsymbol{c}, \boldsymbol{v}, \boldsymbol{x}_{ini}, \boldsymbol{u}_{ini}\}$ and $\{\boldsymbol{c}, \boldsymbol{v}, \boldsymbol{x}_{new}, \boldsymbol{u}_{new}\}$ respectively, and
        the values are denoted as $a_{ini}^c$ and $a_{new}^c$.
        $\{\boldsymbol{a}_{opt}, \boldsymbol{u}_{opt}\} \leftarrow MAC(\{\boldsymbol{a}_{new}, \boldsymbol{u}_{new}\}, \{\boldsymbol{a}_{ini}, \boldsymbol{u}_{ini}\}, T)$.
    $T = T \times \alpha^p$.
return $\boldsymbol{x}_{opt}, \boldsymbol{u}_{opt}$.
```

Algorithm 5: LS-Based Alternative Optimal Algorithm

```
Input: Total iterative number $n_t$.
Output: The optimal solution of Problem 1.
// Initialization:
The iterative number $n = 0$, the initial cluster head selection $\boldsymbol{c}$,
the initial virtual node selection $\boldsymbol{v}$, the initial IUAV trajectory
$\boldsymbol{x}, \boldsymbol{u}$;
// Iteration:
while $n \le n_t$ do
    For fixed $\boldsymbol{v}_n$ and $\boldsymbol{x}_n, \boldsymbol{u}_n$, obtain the optimal solution $\boldsymbol{c}_{n+1}$
    of (17) by using Algorithm 1;
    For fixed $\boldsymbol{c}_{n+1}$ and $\boldsymbol{x}_n, \boldsymbol{u}_n$, obtain the optimal solution $\boldsymbol{v}_{n+1}$
    of (19) by using Algorithm 2;
    For fixed $\boldsymbol{c}_{n+1}$ and $\boldsymbol{v}_{n+1}$, obtain the optimal solution
    $\boldsymbol{x}_{n+1}, \boldsymbol{u}_{n+1}$ of (21) by using Algorithm 3;
    $n = n + 1$;
return $\boldsymbol{c}_{opt}, \boldsymbol{v}_{opt}, \boldsymbol{x}_{opt}, \boldsymbol{u}_{opt}$.
```

Proof: The Markov characteristics of the LS-CHS, LS-VS, and LS-TP algorithms are determined by the operators, which are, respectively

$$\Phi_c\left(c_{i,j}\right) \in \mathcal{D}_{i,j}^c \tag{23}$$

$$\Phi_v\left(v_{i,j}^k\right) \in \mathcal{D}_k^v \tag{24}$$

$$\Phi_t(\boldsymbol{o}_i) \in \mathcal{D}_i^t. \tag{25}$$

Equation (23) means that the next cluster head is selected from the neighbor set $\mathcal{D}_{i,j}^c$ constructed by the current cluster head $c_{i,j}$ in the form of Delaunay triangulation, and the result is only related to the current state $c_{i,j}$. Equation (24) indicates that the next virtual node is selected from the neighbor set $\mathcal{D}_k^v$ constructed by the four-adjacency of the current virtual node $v_{i,j}^k$, and the result is only related to the current state $v_{i,j}^k$. Equation (25) indicates that the next cluster head access sequence is selected from the set of results $\mathcal{D}_i^t$ of the current cluster head access sequence $\boldsymbol{o}_i$ perturbed in a 2-opt manner, and the result is only related to the current state $\boldsymbol{o}_i$. Therefore, this state transition of the LS-CHS, LS-VS, and LS-TP algorithms has Markovian properties. In addition, for each iteration parameter δ, only when the solution satisfies the Metropolis sampling stable criteria, the iteration parameter δ of the LS-CHS, LS-VS, and LS-TP algorithms will decrease, so the algorithm is a homogeneous algorithm. ■

Theorem 2: The Markov chains corresponding to the LS-CHS, LS-VS, and LS-TP algorithms are irreducible.

Proof: First, use $\mathcal{D}^c$, $\mathcal{D}^v$, and $\mathcal{D}^t$ to represent the state space of the LS-CHS, LS-VS, and LS-TP algorithm solutions, respectively, and use k_i^j to represent the state of the ith solution in the jth set. For simplicity, use k_i to represent the state of each algorithm in the ith solution, and use D to represent the solution space corresponding to the state. So for $\forall k_i^x, k_j^x \in \mathcal{D}^x, x \in \{c, v, t\}$, we know that $\exists n \ge 1$ makes

$$\begin{aligned}
&\left(P^n\right)_{ij|\delta=\delta_0} \\
&= \sum_{k_1\in\mathcal{D}}\sum_{k_2\in\mathcal{D}},\dots\sum_{k_{n-1}\in\mathcal{D}} P_{ik_1|\delta=\delta_0}P_{k_1k_2|\delta=\delta_0},\dots P_{k_{n-1}f|\delta=\delta_0} \\
&\ge P_{il_1|\delta=\delta_0}P_{l_1l_2|\delta=\delta_0},\dots P_{l_{n-1}j|\delta=\delta_0} \\
&= G_{il_1}A_{il_1|\delta=\delta_0}G_{l_1l_2}A_{l_1l_2|\delta=\delta_0},\dots G_{l_{n-1}j}A_{l_{n-1}j|\delta=\delta_0} \\
&> 0.
\end{aligned} \tag{26}$$

So for $\forall k_i^x, k_j^x \in \mathcal{D}^x, x \in \{c, v, t\}, \exists n \ge 1, \exists l_0, l_1, \dots, l_n \in \mathcal{D}(l_0 = i \wedge l_n = j)$ makes $(P^n)_{ii|\delta=\delta_0} > 0$, so the Markov chains corresponding to the LS-CHS, LS-VS, and LS-TP algorithms are irreducible. ■

Theorem 3: The Markov chains corresponding to the LS-CHS, LS-VS, and LS-TP algorithms are aperiodic.

Proof: The Metropolis acceptance criterion used in this article is in the form of $A(\Delta, \delta) = e^{-\Delta/\delta}$, where Δ is the difference of the solution change, it can be found that $\forall i, j \in \mathcal{D}$, all have $A_{ij} \le 1$, so

$$\begin{aligned}
P_{ii|\delta=\delta_0} &= 1 - \sum_{l\in\mathcal{D}, l\neq i} P_{il|\delta=\delta_0} \\
&= 1 - \sum_{l\in\mathcal{D}, l\neq i} G_{il}A_{il|\delta=\delta_0} \\
&= 1 - \sum_{l\in\mathcal{D}, l\neq i, l\neq j} G_{il}A_{il|\delta=\delta_0} - G_{ij}A_{ij|\delta=\delta_0} \\
&> 1 - \sum_{l\in\mathcal{D}, l\neq i, l\neq j} G_{il} - G_{ij} \\
&= 1 - \sum_{l\in\mathcal{D}, l\neq i} G_{il} \\
&= 0.
\end{aligned} \tag{27}$$

So, the Markov chains corresponding to the LS-CHS, LS-VS, and LS-TP algorithms are aperiodic. ■

Theorem 4: The LS-CHS, LS-VS, and LS-TP algorithms are convergent.

Proof: The state-limited, unreduced, and nonperiodic Markov chain has a unique smooth distribution. According to Theorems 2 and 3, the Markov chains related to the LS-CHS, LS-VS, and LS-TP algorithms are irreducible and aperiodic, and the Markov chains are known from the operator constraints are state-limited, so the Markov chains have a stationary distribution on each δ, the LS-CHS, LS-VS, and LS-TP algorithms are convergent. ■

F. TinyML-Based Virtual Node Adjustment Module

When the IUAV flies to the target area for data collection, it may encounter the following situation: the preplanned virtual node is unreachable (such as poor weather conditions or poor geographical environment), so the IUAV needs to adjust the position of the virtual node online. Since the IUAV can be considered an embedded device, it has limited storage space and limited computing power. In this way, it may not be possible to store all the risk map data of the target area on the IUAV[3] or to provide enough computing power to give the adjusted virtual node position in real time. Based on this consideration, we provide a TinyML-based Virtual Node Adjustment Module for IUAV, so as to calculate the optimal adjusted virtual node position without prestoring all the additional risk map data, while meeting the real-time requirements of IUAV.

1) Neural Network Architecture: The neural network architecture of the Virtual Node Adjustment Module is a simple multilayer perceptron structure, where *ReLU()* is used as the activation function and *Sigmoid()* maps the value range to [0, 1].

2) Loss Function and Model Training: The goal of model training is to enable the IUAV to find the optimal hovering point around it at its current location. Since the safety of a path is related to the location of the starting point, end point, and virtual node, calculating the degree of safety a certain hover point brings to the path requires the location information of the starting point, end point, and virtual node. We prepare the data set in advance by randomly selecting nodes, that is, randomly selecting the starting point, end point, and virtual nodes on the risk map in advance, and calculating the normalized safety of the path. Then, each item in the data set is composed of $(\boldsymbol{b}_i, \boldsymbol{e}_i, \boldsymbol{v}_i, r_i)$, where $\boldsymbol{b}_i$ is the starting point position, $\boldsymbol{e}_i$ is the end position, $\boldsymbol{v}_i$ is the virtual node position, and r_i is the normalized risk of the IUAV path from $\boldsymbol{b}_i$ to $\boldsymbol{v}_i$ and then to $\boldsymbol{e}_i$. Then, we use the Virtual Node Adjustment Module to enable IUAV to infer path normalized risks through $\boldsymbol{b}_i, \boldsymbol{e}_i$, and $\boldsymbol{v}_i$ without knowing the risk map, so that we can traverse the optimal virtual node location. When training neural networks, the MSE loss function is used, that is

$$\mathcal{L}_{MSE}^{v}(\boldsymbol{b}_i, \boldsymbol{e}_i, \boldsymbol{v}_i) = \frac{1}{m}\sum_{i=1}^{m}\left(r_i - \hat{r}_i\right)^2. \tag{28}$$

The Adam optimizer is used for training. The learning curves of different learning rates are shown in Fig. 4(a). It can be seen that the effect is better when the learning rate is 0.001. It can be seen from Fig. 4(b) that the prediction performance of the learned model is better, and the true value and the predicted value are near the midline.

V. Performance Analysis

In this section, we will implement the proposed algorithm and compare it with the baseline. The algorithms for comparison are as follows.

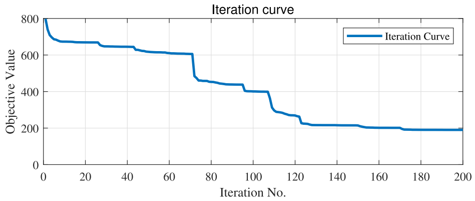


Fig. 3. Iteration curve.

TABLE II
Value of the Simulation Parameters

Notation	Value	Notation	Value
α, β, γ	0.1, 0.8, 0.1	δ_s^c	5×10^3
B	$1MHz$	δ_s^v	5×10^5
N_0	-174dBm/Hz	δ_s^p	5×10^3
$P_{CH_i^t}$	21dBm/Hz	δ_e^c	1×10^{-4}
$P_{i,LoS}$	1dB	δ_e^v	1×10^{-3}
$P_{i,NLoS}$	20dB	δ_e^p	1×10^{-10}
α, β	0.03, 10	L_c	5
H	50m	L_v	2
v_{IUAV}	15m/s	L_p	30
m_{IUAV}	0.5kg	α_c	0.95
r_p	0.2m	α_v	0.9
n_p	4	α_p	0.9
P_{max}	5W	P_c	0.0126W
P_{idle}	0W		

1) *LS-VCT (Proposed):* Based on the local search algorithm, cluster heads selection, IUAV path planning and virtual nodes selection are jointly optimized. By introducing virtual nodes, the risk of the flight path of the IUAV is greatly reduced.
2) *Random-VCT (Proposed):* Adding random selected virtual nodes after jointly optimizing the IUAV path and cluster heads.
3) *DRL-CT [28]:* Based on the deep reinforcement learning, the IUAV flight trajectory and cluster head selection are jointly optimized, but the impact of the risk environment on IUAV flight safety is ignored.

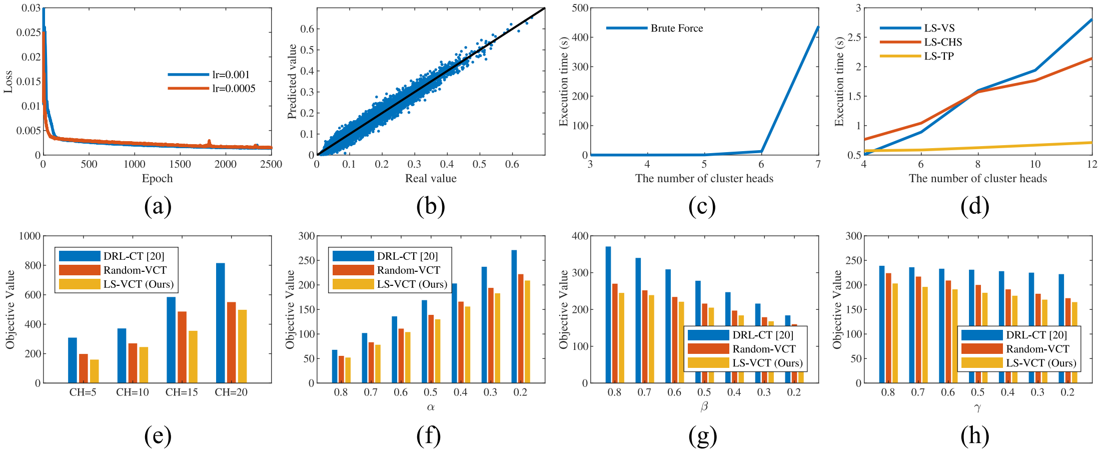
A. Simulation Setup

The size of the IoT network used in the simulation experiment is 1000 m × 1000 m. There are 5, 10, 15, and 20 clusters in the network, and each cluster has 15, 20, 25, and 30 IoT devices, which are randomly deployed in the network. See Table II for other major parameters. The operating system environment of the simulation experiment is Windows 10, the CPU is Intel Core i5-6400T, the main frequency is 2.20 GHz, and the programming language version is Python 3.8.

B. Convergence

First, we tested the convergence of LS-VCT. The test results are shown in Fig. 3. It can be seen that as the iteration proceeds, the objective function value gradually decreases. When it is close to stopping the iteration, the objective function value has almost stopped changing, indicating that it has converged at this time. The experimental results show that the proposed LS-VCT has convergence properties.

[3] A rectangular area with a side length of 2 km, represented by a risk map with a step size of 0.1 m requires storage space of $(2000/0.1)^2 \times 4/1024^3 =$ 1.49 GB.



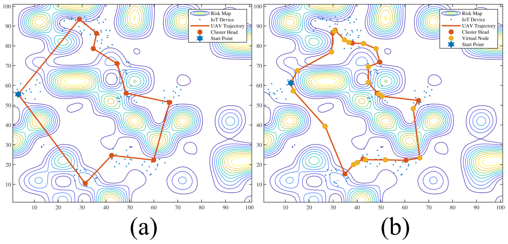
(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 4. Experimental results. (a) Learning curve. (b) Prediction accuracy. (c) Execution time of brute force. (d) Execution time of LS-VTC. (e) Impact of the number of cluster heads. (f) Impact of α. (g) Impact of β. (h) Impact of γ.

TABLE III
ABLATION STUDY RESULTS

LS-VS	LS-TP	LS-CHS	Optimization Results
✗	✗	✗	1445.6881
✓	✗	✗	641.7801
✗	✓	✗	560.2061
✗	✗	✓	503.7879
✓	✓	✗	267.9993
✓	✓	✓	194.4875



(a) (b)

Fig. 5. Changes in IUAV trajectory and cluster heads.

C. Ablation Study

Our proposed LS-VCT has three modules, namely, LS-TP, LS-VS, and LS-CHS. In order to explore the impact of each module on the final result, we removed the corresponding modules and conducted multiple comparative experiments. The experimental results are shown in Table III. It can be seen from Table III that the three modules of LS-VCT proposed in this article all have an optimization effect, and the combined use can reduce the cost by up to 86.55%.

D. Comparison of Execution Time of Algorithms

In this section, we give the execution time of the brute-force algorithm in the experiment, as well as the execution time of each of the three models of LS-VCT. The experimental results are shown in Fig. 4(c) and (d).

From Fig. 4(c), as the number of cluster heads in the network increases, the execution time of the brute force algorithm also increases, and the increase is larger than exponential. When the number of cluster head nodes in the network is 7, the execution time of the brute-force algorithm has exceeded 400 s. However Fig. 4(d) indicates that When the number of cluster head nodes in the network is 7, the execution time of the three models of LS-VTC does not exceed 2 s. Therefore, it can be concluded that the algorithm proposed in this article significantly reduces the execution time compared to the exact algorithm.

E. Impact of Number of Cluster Heads

We can see from Fig. 4(e) the impact of changes in the number of cluster heads in the network on the performance of each algorithm. It can be seen that when the number of cluster heads in the network increases, the flight trajectory of the IUAV becomes more complex, resulting in an upward trend in the value of the objective function. A horizontal comparison of the three algorithms shows that the algorithm proposed in this article has a smaller target value than the baseline algorithm, indicating that the proposed algorithm has better performance. Numerically, LS-VCT reduces the cost by 19.6% on average compared to Random-VCT and 65.4% on average compared with DRL-CT.

F. Impact of Weights in Objective Function

1) Varying α: In order to compare the impact of changes in α on the results, we change α within the range of [0.2, 0.8], and the remaining parameters are set to $(1-\alpha/2)$, The results are shown in Fig. 4(f). It can be seen from Fig. 4(f) that as the weight α decreases, the objective function value increases, and the algorithm proposed has a smaller objective value than the baseline algorithm. Numerically, LS-VCT reduces the cost by 6.34% on average compared to Random-VCT and 29.98% on average compared with DRL-CT.

2) Varying β: In order to compare the impact of changes in β on the results, we change β within the range of [0.2, 0.8], and the remaining parameters are set to $(1-\beta/2)$, the results are shown in Fig. 4(g). It can be seen from Fig. 4(g) that as the weight β decreases, the objective function value decreases, and the algorithm proposed has a smaller objective value than the baseline algorithm. Numerically, LS-VCT reduces the cost

TABLE IV
INFERENCE SPEED OF TINYML-BASED VIRTUAL NODE ADJUSTMENT MODULE

	$\tau = 2$	$\tau = 1$	$\tau = 0.5$
$L = 10m$	0.00300s	0.01101s	0.04100s
$L = 20m$	0.01800s	0.04201s	0.20104s
$L = 40m$	0.04201s	0.20404s	0.90220s

by 6.96% on average compared to Random-VCT and 37.94% on average compared with DRL-CT.

3) Varying γ: In order to compare the impact of changes in γ on the results, we change γ within the range of [0.2, 0.8], and the remaining parameters are set to $(1 - \gamma/2)$, The results are shown in Fig. 4(h). It can be seen from Fig. 4(h) that as the weight γ decreases, the objective function value increases, and the algorithm proposed has a smaller objective value than the baseline algorithm. Numerically, LS-VCT reduces the cost by 8.47% on average compared to Random-VCT and 25.47% on average compared with DRL-CT. We visualized the changes of UAV trajectory and cluster heads caused by LS-VCT. The visualization results are shown in Fig. 5.

G. Real-Time Inference Performance

The TinyML-based Virtual Node Adjustment Module should meet the real-time requirements of IUAVs in order to find high-quality virtual nodes for IUAVs in emergencies. For such a TinyML-based application, its real-time performance is reflected in the inference speed. We tested the inference speed under different input scales, and the results are shown in Table IV. It can be seen that when the IUAV needs to adjust a virtual node in an emergency, the time required to calculate the optimal virtual node position will not exceed 1 s, indicating that it has good real-time performance.

H. Dynamic Quantification of Model

Since the Virtual Node Adjustment Module has a Linear layer, the size of the space occupied by the model can be reduced through the quantification method of the model, allowing it to be deployed on a wider range of embedded devices. For this, we use pytorch's *torch.quantization.quantize_dynamic()* function to quantify the model. The size of the model before quantization is 3.79 kB, and the size of the model after quantization is 1.39 kB, which reduces the storage space occupied by nearly three times.

VI. CONCLUSION

In this article, we propose a UAV path-planning algorithm based on local search and TinyML for joint optimization of energy consumption and risk in IoT networks. In order to reduce the complexity of solving the problem, we split the problem into three subproblems, namely, cluster head selection problem, virtual node selection problem, and node access sequence problem. Then use the local search algorithm to solve each subproblem, and then use the iterative optimization method to solve the above three subproblems in sequence until convergence. At the same time, considering that IUAV may encounter unexpected situations when performing data collection tasks and need to adjust the location of virtual nodes in real time, we proposed a TinyML framework to assist IUAV in reasoning about the location of the optimal virtual node. And the dynamic quantization method is used to reduce the storage space occupied by the model. Finally, a large number of simulation experiments show that the proposed algorithm is effective and its performance exceeds the baseline algorithm.

In the future, joint optimization of energy consumption and risk for IUAVs trajectory in a 3-D environment can be studied which is also our future research direction. At the same time, it is of practical significance that we can also consider designing different risk factor calculation models to adapt to the needs of different environments.

REFERENCES

[1] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2624–2661, 4th Quart., 2016.

[2] Z. Ning et al., "5G-enabled UAV-to-community offloading: Joint trajectory design and task scheduling," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3306–3320, Nov. 2021.

[3] R. Yang, J. Ma, J. Zhang, S. Kumari, S. Kumar, and J. J. P. C. Rodrigues, "Practical feature inference attack in vertical federated learning during prediction in artificial Internet of Things," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 5–16, Jan. 2024.

[4] J. Bai, J. Gui, G. Huang, S. Zhang, and A. Liu, "UAV-supported intelligent truth discovery to achieve low-cost communications in mobile crowd sensing," *Digit. Commun. Netw.*, to be published.

[5] M. Dai, Z. Su, Q. Xu, Y. Wang, and N. Lu, "A trust-driven contract incentive scheme for mobile crowd-sensing networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 1794–1806, Feb. 2022.

[6] D. Wei, J. Zhang, M. Shojafar, S. Kumari, N. Xi, and J. Ma, "Privacy-aware multiagent deep reinforcement learning for task offloading in VANET," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 13108–13122, Nov. 2023.

[7] Z. Ning, Y. Yang, X. Wang, Q. Song, L. Guo, and A. Jamalipour, "Multi-agent deep reinforcement learning based UAV trajectory optimization for differentiated services," *IEEE Trans. Mobile Comput.*, early access, Sep. 5, 2023, doi: 10.1109/TMC.2023.3312276.

[8] Z. Ning et al., "Dynamic computation offloading and server deployment for UAV-enabled multi-access edge computing," *IEEE Trans. Mobile Comput.*, vol. 22, no. 5, pp. 2628–2644, May 2023.

[9] H. Dai et al., "Bloom filter with noisy coding framework for multi-set membership testing," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 6710–6724, Jul. 2023.

[10] Z. Ning, H. Chen, E. C. Ngai, X. Wang, L. Guo, and J. Liu, "Lightweight imitation learning for real-time cooperative service migration," *IEEE Trans. Mobile Comput.*, early access, Jan. 25, 2023, doi: 10.1109/TMC.2023.3239845.

[11] X. Wang, Z. Ning, S. Guo, M. Wen, L. Guo, and H. V. Poor, "Dynamic UAV deployment for differentiated services: A multi-agent imitation learning based approach," *IEEE Trans. Mobile Comput.*, vol. 22, no. 4, pp. 2131–2146, Apr. 2023.

[12] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in UAV enabled wireless sensor network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 328–331, Jun. 2018.

[13] R. Liu et al., "DRL-UTPS: DRL-based trajectory planning for unmanned aerial vehicles for data collection in dynamic IoT network," *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1204–1218, Feb. 2023.

[14] W. Jin, J. Yang, Y. Fang, and W. Feng, "Research on application and deployment of UAV in emergency response," in *Proc. IEEE 10th Int. Conf. Electron. Inf. Emerg. Commun. (ICEIEC)*, 2020, pp. 277–280.

[15] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen, and A. Ghrayeb, "UAV trajectory planning for data collection from time-constrained IoT devices," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 34–46, Jan. 2020.

[16] J. Liu, P. Tong, X. Wang, B. Bai, and H. Dai, "UAV-aided data collection for information freshness in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2368–2382, Apr. 2021.

[17] Y. Zhou, X. Ma, S. Hu, D. Zhou, N. Cheng, and N. Lu, "QoE-driven adaptive deployment strategy of multi-UAV networks based on hybrid deep reinforcement learning," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5868–5881, Apr. 2022.

[18] R. Liu, A. Liu, Z. Qu, and N. N. Xiong, "An UAV-enabled intelligent connected transportation system with 6G communications for Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2045–2059, Feb. 2023.

[19] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and J. Henry, "UAV trajectory planning in wireless sensor networks for energy consumption minimization by deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9540–9554, Sep. 2021.

[20] J. Bai, G. Huang, S. Zhang, Z. Zeng, and A. Liu, "GA-DCTSP: An intelligent active data processing scheme for UAV-enabled edge computing," *IEEE Internet Things J.*, vol. 10, no. 6, pp. 4891–4906, Mar. 2023.

[21] L. Quan, Z. Zhang, X. Zhong, C. Xu, and F. Gao, "EVA-planner: Environmental adaptive quadrotor planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 398–404.

[22] X. Fu, A. Liu, N. N. Xiong, T. Wang, and S. Zhang, "ATWR-SMR: An area-constrained truthful worker recruitment based sensing map recovery scheme for sparse MCS in extreme-environment Internet-of-Things," *IEEE Internet Things J.*, early access, Sep. 12, 2023, doi: 10.1109/JIOT.2023.3314615.

[23] A. Choudhry, B. Moon, J. Patrikar, C. Samaras, and S. Scherer, "CVaR-based flight energy risk assessment for multirotor UAVs using a deep energy model," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 262–268.

[24] J. Bai, Z. Zeng, T. Wang, S. Zhang, N. N. Xiong, and A. Liu, "TANTO: An effective trust-based unmanned aerial vehicle computing system for the Internet of Things," *IEEE Internet Things J.*, vol. 10, no. 7, pp. 5644–5661, Apr. 2023.

[25] H. Dai, K. Sun, A. X. Liu, L. Zhang, J. Zheng, and G. Chen, "Charging task scheduling for directional wireless charger networks," *IEEE Trans. Mobile Comput.*, vol. 20, no. 11, pp. 3163–3180, Nov. 2021.

[26] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annu. Hawaii Int. Conf. System Sci.*, 2000, p. 10.

[27] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 366–379, Oct.–Dec. 2004.

[28] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and J. Henry, "Joint cluster head selection and trajectory planning in UAV-aided IoT networks by reinforcement learning with sequential model," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12071–12084, Jul. 2022.

[29] D. Ebrahimi, S. Sharafeddine, P.-H. Ho, and C. Assi, "Data collection in wireless sensor networks using UAV and compressive data gathering," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2018, pp. 1–7.

[30] W. Feng et al., "Joint 3D trajectory and power optimization for UAV-aided mmWave MIMO-NOMA networks," *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2346–2358, Apr. 2021.

[31] J. Feng and J. Gong, "AoI-optimal data collection, offloading, and migration in mobile edge networks," in *Proc. IEEE 24th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, 2023, pp. 147–156.

[32] B. Miller, K. Stepanyan, A. Miller, and M. Andreev, "3D path planning in a threat environment," in *Proc. 50th IEEE Conf. Decis. Control Eur. Control Conf.*, 2011, pp. 6864–6869.

[33] Y. Qu, Y. Zhang, and Y. Zhang, "Optimal flight path planning for UAVs in 3-D threat environment," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, 2014, pp. 149–155.

[34] N. Wen, L. Zhao, X. Su, and P. Ma, "UAV online path planning algorithm in a low altitude dangerous environment," *IEEE/CAA J. Automatica Sinica*, vol. 2, no. 2, pp. 173–185, Apr. 2015.

[35] M. Giordano, N. Baumann, M. Crabolu, R. Fischer, G. Bellusci, and M. Magno, "Design and performance evaluation of an Ultralow-power smart IoT device with embedded TinyML for asset activity monitoring," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, Apr. 2022. [Online]. Available: https://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=9717300&punumber=19

[36] M. Pavan, A. Caltabiano, and M. Roveri, "TinyML for UWB-radar based presence detection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2022, pp. 1–8.

[37] S. Lim, H. Yu, and H. Lee, "Optimal tethered-UAV deployment in A2G communication networks: Multi-agent Q-learning approach," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18539–18549, Oct. 2022.

[38] S. Lee and H. S. Lee, "Analysis of network lifetime in cluster-based sensor networks," *IEEE Commun. Lett.*, vol. 14, no. 10, pp. 900–902, Oct. 2010.

[39] H. Al-Hamadi and I.-R. Chen, "Redundancy management of multipath routing for intrusion tolerance in heterogeneous wireless sensor networks," *IEEE Trans. Netw. Service Manag.*, vol. 10, no. 2, pp. 189–203, Jun. 2013.

[40] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *J. ACM*, vol. 7, no. 4, pp. 326–329, 1960.

[41] I. O. Bohachevsky, M. E. Johnson, and M. L. Stein, "Generalized simulated annealing for function optimization," *Technometrics*, vol. 28, no. 3, pp. 209–217, 1986.

Run Liu (Graduate Student Member, IEEE) received the B.E. degree in Internet of Things engineering from the School of Computer Science and Engineering, Central South University, Changsha, China, in 2022.

His research interests include wireless networks, edge computing, and deep reinforcement learning.

Mande Xie was born in 1977. He received the Ph.D. degree in circuit and system from Zhejiang University, Hangzhou, China, in 2006.

He is currently a Professor with Zhejiang Gongshang University, Hangzhou. His research interests include wireless sensor networks, social network, and privacy preservation.

Anfeng Liu received the M.Sc. and Ph.D. degrees in computer science from Central South University, Changsha, China, in 2002 and 2005, respectively.

He is currently a Professor with the School of Information Science and Engineering, Central South University. His major research interests include wireless sensor networks, Internet of Things, information security, edge computing, and crowdsensing.

Houbing Song (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, VA, USA, in August 2012.

He is currently a Tenured Associate Professor, the Director of the NSF Center for Aviation Big Data Analytics (Planning), the Associate Director for Leadership of the DOT Transportation Cybersecurity Center for Advanced Research and Education (Tier 1 Center), and the Director of the Security and Optimization for Networked Globe Laboratory (www.SONGLab.us), University of Maryland at Baltimore County, Baltimore, MD, USA.

Dr. Song was a recipient of over ten best paper awards from major international conferences. He is an ACM Distinguished Member.