Path Planning and Control of Intelligent Delivery UAV Based on Internet of Things and Edge Computing

Xiuzhu Zhang
School of Economics and Management, Jiaozuo University, Jiaozuo China

***Abstract*—This paper investigates the intelligent delivery UAV path planning and control problem based on the Internet of Things and edge computing, and proposes a novel model and algorithm to realize the collaborative optimization of the path planning and control of the UAV, which improves the intelligence level and flight efficiency of the UAV. In this paper, the mathematical model of UAV path planning and control is firstly established, the relationship and influencing factors among the elements of UAV, edge server, delivery task, path planning and control are analyzed, and the optimization objectives and constraints are proposed. Then, this paper designs an algorithmic framework for UAV path planning and control, using the support and guidance of edge computing to achieve the cooperative optimization of path planning and control of UAVs, taking into account the constraints and objectives of the UAVs themselves, as well as the synergy and competition between UAVs. Then, this paper proposes specific algorithms for UAV path planning and control, adopting methods such as meta-heuristics, to solve the optimization problem of UAV path planning and control, and improve the intelligent level and flight performance of UAVs.**

***Keywords*—Internet of things; edge computing; smart distribution; drone path; planning and control**

I. Introduction

With the development of IoT technology, more and more smart devices are connected to the Internet, forming a huge data source. This data is characterized by massive, diverse, real-time, dynamic, etc., which pose great challenges to traditional cloud computing platforms, such as high latency, low bandwidth, low reliability, and high energy consumption. To address these issues, edge computing, as an emerging computing paradigm, shifts computing resources from cloud centers to servers on the edge side of the network to provide computing support for connected end devices. Edge computing enables local processing and analysis of data, reduces data transmission and latency, and improves the quality of service and user experience while protecting data security and privacy [1].

The computing power and battery capacity of UAVs are very limited, which cannot meet the demands of complex data processing and long flight times. Therefore, combining edge computing and UAVs to build a mobile edge computing network based on UAVs is an effective solution. In this kind of network, UAVs can transmit data to the edge side for fast processing and analysis through the communication connection with edge servers, and at the same time obtain guidance and support from the edge side to improve the intelligence level and flight efficiency of UAVs. The generalized UAV path planning framework is shown in Fig. 1 [2], [3].

Path planning and control of UAVs is one of the core technologies of UAVs, which determines the flight trajectory and maneuvers of UAVs and directly affects the performance and safety of UAVs. In the UAV-based mobile edge computing network, the path planning and control of UAVs should not only consider the UAVs' own constraints and objectives, such as flight time, energy consumption, load, and task completion, but also consider the influencing factors of edge computing, such as the location and number of edge servers, computational capacity, and communication link quality. In addition, when multiple UAVs perform delivery tasks in the same area at the same time, the synergy and competition between UAVs, such as path conflict, resource allocation, and task coordination, are also considered. Therefore, the study of intelligent delivery UAV path planning and control based on IoT and edge computing is a topic of great theoretical significance and practical value [4].

Current research has mainly focused on the enhancement of quality of service and user experience of IoT applications by edge computing, while less consideration has been given to the enhancement of intelligence and efficiency of IoT devices by edge computing. As a typical IoT device, the improvement of intelligence and efficiency of UAVs can not only enhance the function and performance of UAVs, but also reduce the operation cost and risk of UAVs [5]. Therefore, it is a meaningful work to study how to optimize the path planning and control of UAVs using edge computing.

Under the cooperative operation environment, the edge computing-based intelligent delivery UAV system can better balance the load of each UAV and ensure the fairness of task allocation and the effectiveness of resource use by dynamically adjusting the path planning strategy. In addition, this approach can also achieve dynamic optimization of UAV paths, avoid flight conflicts, and adapt to changing environmental conditions and task priorities, thus significantly improving the operational efficiency and task completion quality of the entire UAV cluster.

Against the background of the current rapid development of the Internet of Things (IoT) and drone technology, intelligent distribution drones have shown broad application prospects in the field of logistics and distribution due to their unique advantages of high efficiency, convenience and flexibility. However, in the process of executing distribution tasks, UAVs are limited by their own limited computing power, short battery life and instability of wireless communication, especially in the face of large data volume, real-time response requirements of high scenarios, the traditional centralized cloud computing architecture is difficult to meet the needs of efficient and accurate path planning and control. In addition, when multiple UAVs work together, it is also necessary to take into account the fairness of task allocation, path conflict avoidance, and overall task completion efficiency and many other issues [6], [7]. Therefore, this paper proposes a research method for intelligent distribution UAV path planning and control based on IoT and edge computing, which can make up for the shortcomings of traditional methods in processing large-scale data, real-time decision-making, and responding to changes in the local environment of the UAV, and by integrating the advantages of IoT and edge computing, it can effectively solve the computational bottlenecks and communication delays faced by UAVs when they perform their tasks. IoT technology enables UAVs to acquire and transmit rich environmental information in real time, while edge computing can provide instant computing resources and services near the data source, greatly reducing the delay of data transmission and improving the data processing speed, which in turn supports UAVs to make more accurate and real-time path planning and control decisions.

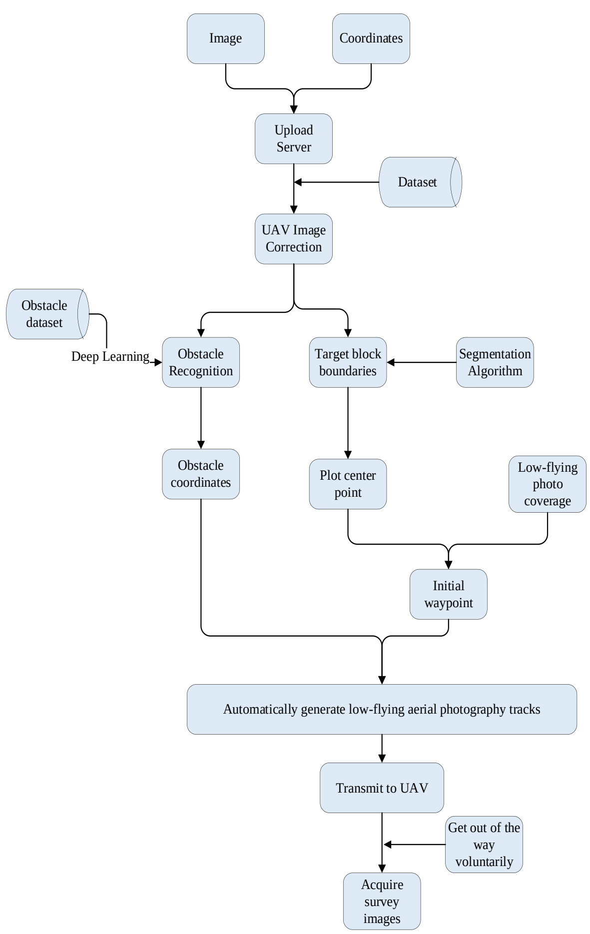

Fig. 1. Drone path planning model.

The research contributions of this paper mainly include the following aspects: (1) Constructed a mathematical model for intelligent delivery UAV path planning and control based on IoT and edge computing environment, deeply analyzed the interaction mechanism among core elements such as UAVs, edge servers, and delivery tasks, and clarified the objective function and necessary constraints for optimizing the path planning and control of UAVs. (2) Designed a set of new Algorithmic framework, which takes into account the individual performance constraints and target requirements of a single UAV when performing a task, as well as the fair scheduling, path conflict avoidance, and overall task efficiency optimization of multiple UAVs when performing tasks together. (3) A specific path planning and control algorithm for intelligent delivery UAVs is proposed, which combines the heuristic, meta-heuristic, and machine learning methods to solve the problems of UAVs in actual operation. The path optimization problem of UAVs in actual operation improves the intelligence level and flight execution efficiency of the UAV system.

The main content is divided into five sections. Section I introduces the research background. Section II analyzes the current research status. Section III explains the research methods. Section IV analyzes and discusses the research results. Finally, Section V concludes the paper..

II. LITERATURE REVIEW

Cui et al. [8] studied a cooperative path planning algorithm for UAV clusters based on edge computing, which uses the location information and movement laws of the edge server to guide the path planning of the UAVs, while taking into account the synergy and competition between the UAVs, to optimize the UAVs' task completion and flight efficiency. Cui et al. [9] investigated a collaborative task allocation method for UAV clusters based on edge computing, which utilizes the edge server's computational capability and data analysis advantages to provide decision support for task allocation for UAV clusters, while optimizing the UAVs' task execution effectiveness by considering the UAVs' energy consumption, flight time, and task priority. Dec et al. [10] utilized the communication resources and link quality of the edge server to provide communication optimization for UAV clusters, while factors such as communication demand, communication interference and communication cost of UAVs are considered to optimize the communication performance and communication efficiency of UAVs. Ding et al. [11] investigated a cooperative security assurance method for UAV clusters based on edge computing, which utilizes the security technology and security policy of edge servers to provide services for UAV clusters to provide security assurance, while considering factors such as the security demand, security threat and security cost of UAVs, to optimize the level of security and the security benefit of UAVs.

In summary, the research related to UAV path planning and control has been relatively mature, and the number of its results is shown in Fig. 2. Although the above series of studies have made significant progress in edge computing-based UAV cluster path planning, task assignment, autonomous navigation, communication optimization, and security, there are still some important limitations and knowledge gaps to be addressed in this area. First, most current research focuses on single or partial optimization objectives, such as information update speed, energy consumption, flight time, and mission completion, while it remains a challenge to maximize the overall effectiveness of UAV clusters in the context of multi-objective optimization. Second, although edge computing enhances the real-time computation and decision-making capabilities of UAVs, in practical applications, the computational resources of edge servers are not unlimited, and how to effectively schedule and utilize them under resource-constrained conditions to cope with large-scale and high-density UAV cluster operations is an issue that needs to be explored in depth. Furthermore, the reliable communication, obstacle avoidance and adaptive flight capabilities of UAVs in complex and dynamic environments need to be further strengthened, especially in extreme or unexpected situations, how to utilize edge computing technology to improve the UAV's anti-interference capability and fault recovery speed, and to safeguard the flight safety and service continuity needs more research. In addition, current research has not paid enough attention to and explored in-depth the compliance issues of edge computing in UAV applications, which involve user privacy, data security, and regulatory compliance. In summary, although edge computing-based UAV clustering research has achieved a series of results, limitations in multi-objective optimization, efficient scheduling in resource-constrained environments, adaptation to complex environments, and legal and ethical issues reveal the large research space and development potential that remain in this area.

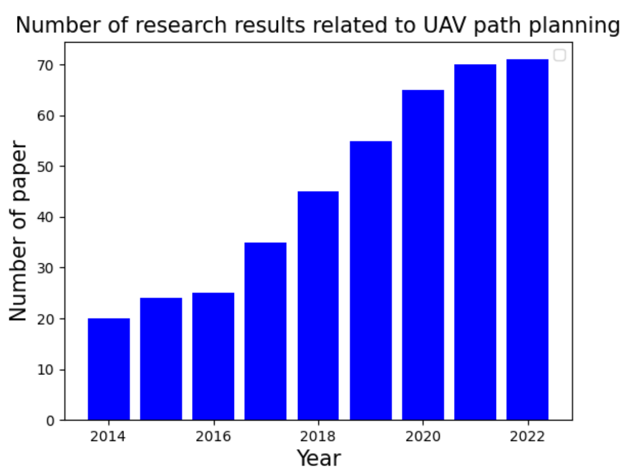

Fig. 2. Number of research results related to UAV path planning.

III. Research Methodology

A. UAV Path Planning Model

Suppose there are N drones, M edge servers, and K delivery tasks. Each UAV i has an initial position p_i^0 and a target position p_i^f, as well as some constraints, such as maximum speed v_{imax}, maximum acceleration a_{imax}, maximum turning angle θ_{imax}, maximum flight time t_{imax}, maximum payload w_{imax} etc. [12]. Each edge server j has a fixed location q_j, as well as some resource parameters such as computing power c_j, storage capacity s_j, communication bandwidth b_j, etc. [13], [14]. Each delivery task k has a weight of the demanded item w_k, a location of the demanded item r_{ks}, a location of the delivery destination r_{kd}, and a delivery time window $[t_{kmin}, t_{kmax}]$. The distance between the UAV, the edge server, and the delivery task can be measured in terms of the Euclidean distance, i.e., $d(x,y)=(x-y)^T(x-y)$, where *x* and *y* are any two position vectors [15], [16].

The purpose of path planning and control of UAVs is to find a set of optimal control inputs $u_i(t)$ that enable the UAV to complete the delivery task while satisfying constraints and minimizing some optimization objective function. The optimization objective function can be the UAVs' total flight time, total flight distance, total energy consumption, total delay, etc., or it can be a weighted function that integrates several factors. For example, if the optimization objective is to minimize the total flight time of the UAV, then the optimization objective function is specifically shown in Eq. (1). where, t_i^f is the time for the UAV i to reach the target position [17], [18].

$$\min \sum_{i=1}^{N} \int_0^{t_i^f} dt \tag{1}$$

The constraints for path planning and control of UAVs include kinematics and dynamics constraints of UAVs, collision avoidance constraints between UAVs, communication connectivity constraints between UAVs and edge servers, matching constraints between UAVs and delivery tasks, and time window constraints for delivery tasks. Each delivery task can only be executed by one UAV, as shown in Eq. (2). Where y_{ik} indicates whether the UAV i executes the delivery task k, which takes the value of 0 or 1. The load capacity of the UAV cannot exceed the maximum limit, i.e., Eq. (3) [19].

$$x_k = \sum_{i=1}^{N} y_{ik}, \quad k = 1, \ldots, K \tag{2}$$

$$\sum_{k=1}^{K} w_{ik} y_{ik} \le w_{imax}, \quad i = 1, \ldots, N \tag{3}$$

where, w_{ik} denotes the weight of the item for the delivery mission k and w_{imax} denotes the maximum load capacity of the drone i.

The drone cannot fly beyond the maximum limit, as shown in Eq. (4) [20].

$$\sum_{k=1}^{K}\sum_{j=1}^{M}(d(s_k,t_k)y_{ik}+d(p_i^0,q_j)z_{ij} +d(q_j,p_i^t)z_{ij})\le d_{imax},\quad i=1,\ldots,N \tag{4}$$

where, $d(\cdot,\cdot)$ denotes the distance between two locations, s_k denotes the location of the demanded item of the delivery task k, t_k denotes the location of the delivery destination of the delivery task k, p_i^0 denotes the initial location of the drone i, q_j denotes the location of the edge server j, p_i^t denotes the location of the drone i, d_{imax} denotes the maximum flight distance of the drone i, and z_{ij}. Indicates whether the drone i is connected to the edge server j, which takes the value of 0 or 1. The drone must complete the delivery within the time window of the delivery task, as shown in Eq. (5) [21], [22].

$$t_k=\max\{y_{ik}=1\}\left(\sum_{j=1}^{M}\frac{d(p_i^0,q_j)}{v_i z_{ij}+L_i z_{ij}}+d(s_k,t_k)\right),\quad k=1,\ldots,K \tag{5}$$

B. Algorithmic Framework

The path planning and control algorithm framework for smart delivery UAVs based on IoT and edge computing builds a complete set of decision-making processes, as shown in Fig. 3. The framework covers six key steps from information interaction to real-time control:

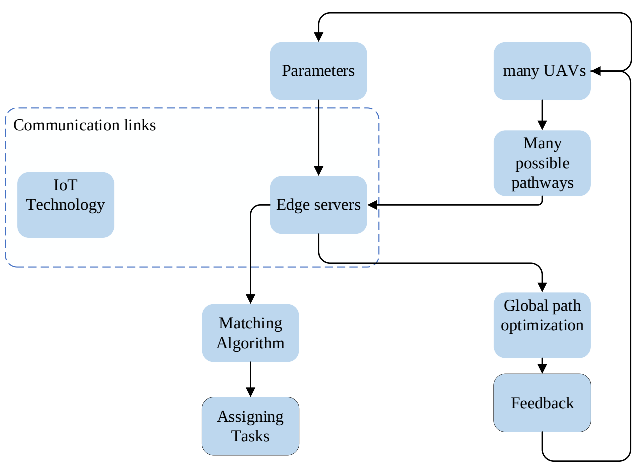

Fig. 3. Path planning and control algorithm framework based on IoT and edge computing.

Step I: In the initialization phase of the system, the UAV establishes a stable communication link with the edge server through IoT technology, and sends its real-time status data as well as the specific demand parameters of the delivery task it is carrying to the edge server. The server receives and stores this information and analyzes and preprocesses it in depth [23].

Step II: The edge server uses matching algorithms to assign optimal or sub-optimal delivery tasks to each UAV based on the received UAV status and task demand information, aiming to achieve an optimal balance of several key performance indicators, such as total flight time, distance, energy consumption, and latency. Subsequently, the server will send the allocation results back to the relevant drones in real time [24].

Step III: Immediately after getting the task assignment, the UAV uses the path planning algorithm to autonomously design an efficient flight path from its current location to the target location based on the matching scheme provided by the edge server. In this process, the kinematics and dynamics constraints of the UAV itself are fully considered to ensure safe flight while avoiding collisions with other UAVs and the time window requirements of the delivery task are strictly followed. After the planning is completed, the UAV sends the finalized path information to the edge server again [25].

Step IV: Based on the path planning results submitted by all UAVs, the edge server performs global path optimization using a cooperative optimization algorithm to promote effective collaboration and competition among multiple UAVs, so as to achieve the overall optimal or near-optimal path layout of the entire distribution network. The optimized path planning scheme is then fed back to the participating drones.

Step V: The UAV applies the appropriate control algorithm to generate a set of best-fit control input commands based on the co-optimization path returned by the server. The UAV performs precise flight operations accordingly and is able to adjust its control strategy in real time to respond to changing environmental factors. This set of control inputs is also reported by the UAV to the edge server for monitoring and recording [26].

Step VI: In the whole monitoring process, the edge server utilizes monitoring algorithms to monitor and estimate the actual flight status of each UAV in real time, and to evaluate and feedback its flight performance, forming a closed-loop control system. Fig. 3 is the algorithm for path planning and control of smart delivery UAVs based on IoT and edge computing [27].

C. Solution Algorithm

The algorithm is divided into two levels, local planning and global optimization, using the collaboration between the edge nodes of the UAV and the cloud to achieve the goal of finding an optimal or near-optimal path that satisfies multiple objective functions in a dynamically changing environment. This study will explain each step of the algorithm step by step below:

Step 1: At the edge node of the UAV, based on the current position, speed, target, obstacles and other information, a local path planning algorithm, such as the artificial potential field method, etc., is used to generate a short-term path, i.e., a genotype X. The specific formula is: $X=(x_1,x_2,\ldots,x_N)$ where N is the length of the genotype, which is determined by the maximal flight time of the UAV, T, and the flight interval, Δt, i.e., $N=T/\Delta t$. x_i is the i th flight action, which takes the value of $\{A, B, L, R, U, D\}$ and means forward, backward, left turn, right turn, up and down, respectively. For example, $X = (A, A, L, U, A, R, D, A, \ldots)$ means the drone first advances two steps, then turns left, rises, advances, turns right, descends, advances, and so on [28].

This study utilize the artificial potential field method to solve the initial solution. This study abstract the operating environment of the UAV as a potential field, in which the target point exerts a gravitational force on the UAV, the obstacle exerts a repulsive force on the UAV, and the UAV, under the action of the combined force, moves in the direction of decreasing potential energy until it reaches the target point or encounters a local minimum. The specific formula is: $F = F_a + F_r$ where F is the combined force, F_a is the gravitational force, and F_r is the repulsive force. The formula for the gravitational force is: $F_a = -k_a \nabla U_a$ where k_a is the gravitational coefficient, ∇U_a is the gradient of the gravitational potential field, and U_a is the gravitational potential field function, which is generally defined as a function of the distance from the UAV to the target point, i.e. $U_a = \frac{1}{2} k_a \rho^2 (X, X_g)$ where $\rho(X, X_g)$ is the Euclidean distance between the position of the UAV X and the position of the target point X_g, i.e. The Euclidean distance, i.e.: $\rho(X, X_g) = \sqrt{(x - x_g)^2 + (y - y_g)^2 + (z - z_g)^2}$ The formula for repulsion is: $F_r = k_r \nabla U_r$ where k_r is the repulsion coefficient, ∇U_r is the gradient of the repulsive potential field, and U_r is the repulsive potential field function, generally defined as a function of the reciprocal of the distance from the UAV to the obstacle [29].

In order to avoid the UAV being affected by obstacles that are too far away, a maximum influence distance of ρ_0 is generally set, and the repulsion force is zero when $\rho(X, X_o) > \rho_0$. In order to avoid the UAV falling into a local minimum, some heuristics are generally set, such as increasing the virtual target point, changing the direction of the repulsion force, and increasing the memory.

Step 2: On the edge node of the UAV, calculate the fitness value of the genotype, i.e., $f(X)$, and communicate it with the edge nodes of other UAVs to exchange information and coordinate conflicts to form a local population Pl. The fitness function is a function used to measure the merit of the genotype, defined as a weighted sum of multiple objective functions, as shown in (6).

$$f(X) = w_1 F_t(X) + w_2 F_d(X) + w_3 F_e(X) + w_4 F_l(X) \tag{6}$$

where, w_1, w_2, w_3, w_4 is the weight coefficient, $F_t(X)$ is the total flight time of the UAV, $F_d(X)$ is the total flight distance of the UAV, $F_e(X)$ is the total energy consumption of the UAV, and $F_l(X)$ is the total delay of the UAV [30]. These subfunctions can be computed based on the flight dynamics model and communication model of the UAV. For example, if this study assume that the flight speed of the UAV is v, the flight time interval is Δt, the energy consumption of the flight maneuver is e_i, and the delay of the flight maneuver is l_i, as in Eq. (7)-(10).

$$F_t(X) = N\Delta t \tag{7}$$

$$F_d(X) = Nv\Delta t \tag{8}$$

$$F_e(X) = \sum_{i=1}^{N} e_i \tag{9}$$

$$F_l(X) = \sum_{i=1}^{N} l_i \tag{10}$$

At the edge nodes, UAVs need to communicate with other UAVs to exchange their genotypes and fitness values, as well as other information such as position, speed, and target. Through communication, UAVs can coordinate their flight maneuvers to avoid collision or conflict with other UAVs. The communication can be broadcast, multicast or unicast, the frequency of the communication can be fixed or dynamic, and the protocol of the communication can be TCP, UDP or others. The effectiveness of communication can be measured by metrics such as communication success rate, communication delay, communication overhead, etc. The purpose of communication is to form a localized population P_l, i.e., a set of genotypes, each of which has an adaptation value indicating its degree of superiority or inferiority in the current environment.

Step 3: In the cloud, based on the genotypes and fitness values sent by the edge nodes of all UAVs, a global path optimization algorithm is used to generate a global population P_g by performing crossover, mutation, and selection operations on the local populations, and send it to the edge nodes of the corresponding UAVs. The purpose of the global path optimization algorithm is to improve the global fitness while ensuring the local fitness, i.e., to meet the individual needs of the UAVs while achieving the collective collaboration of the UAVs and optimizing the overall performance. The parameters of the global path optimization algorithm are determined by N_g, i.e., the global population size.

Suppose the parameters of the network are θ and the output of the network is $Q_\theta(s, a)$ which represents the estimate of the value of Q for taking action a in state s. The true reward is r, the next state is s', the next action is a', and the discount factor is γ. Then the error between the output of the network and the true reward is specified as shown in Eq. (11).

$$\delta = Q_\theta(s,a) - (r + \gamma \max_{a'} Q_\theta(s',a')) \tag{11}$$

Here, $r + \gamma \max_{a'} Q_\theta(s',a')$ is the target Q value that represents the expectation of the maximum cumulative reward that can be obtained after taking action a in state S. This error is also called Temporal Difference Error (TDE) and reflects the gap between the network's estimate and the true reward.

In order to make the output of the network closer to the true reward, this study need to minimize this sum of squares of error as shown in Eq. (12).

$$L(\theta) = \frac{1}{2} \sum_{(s,a,r,s')\in D} \delta^2 \tag{12}$$

Here, D is a batch of experience tuples randomly selected from the experience playback pool, also called a mini-batch (Mini-batch). This Loss Function (Loss Function) reflects the performance of the network, the smaller the better.

To minimize this loss function, this study needs to update the parameters of the network using gradient descent or some other optimization algorithm as shown in Eq. (13).

$$\theta \leftarrow \theta - \alpha \nabla_\theta L(\theta) \tag{13}$$

Here, α is the Learning Rate, which controls the step size of the parameter update, and $\nabla_\theta L(\theta)$ is the Gradient of the loss function with respect to the parameter, which indicates the direction of change of the loss function in the parameter space. By updating the parameters in the opposite direction of the gradient, this study can make the loss function gradually decrease, thus making the output of the network closer to the true reward.

Step 4: Evaluate the network, i.e., use the updated network to generate a new genotype, i.e., a new path, for each UAV, and then compute the fitness value of that genotype, i.e., $f(X)$, and evaluate the network according to the size of the fitness value and select the optimal or better network as the current optimal or near-optimal solution.

Assuming that the parameters of the updated network are θ', for each UAV, this study can use the network to generate a new genotype as shown in Eq. (14).

$$X = (x_1, x_2, \ldots, x_N) \tag{14}$$

where, $x_i = argmax_a Q_{\theta'}(s_i,a)$, denotes the action with the largest value of Q output by the network in the state s_i. This genotype is the output of the network and indicates the optimal or near-optimal path given by the network.

Then, this study can calculate the fitness value for that genotype as shown in Eq. (15).

$$f(X) = w_1 F_t(X) + w_2 F_d(X) + w_3 F_e(X) + w_4 F_l(X) \tag{15}$$

Here, w_1, w_2, w_3, w_4 is the weight coefficient, $F_t(X)$ is the total flight time of the UAV, $F_d(X)$ is the total flight distance of the UAV, $F_e(X)$ is the total energy consumption of the UAV, and $F_l(X)$ is the total delay of the UAV. These subfunctions can be calculated based on the flight dynamics model and communication model of the UAV. This fitness value reflects the merit of the genotype, the larger the better.

Finally, this study can evaluate the networks based on the magnitude of the fitness values and select the optimal or better network as the current optimal or near-optimal solution. For example, this study can use a sliding window to record the parameter and fitness values of a number of recent networks, and then select the network with the largest fitness value from them, or use a Softmax function to randomly select a network based on the proportion of fitness values.

Step 5: Termination judgment, i.e., to determine whether the preset termination conditions, such as the maximum number of training times, the minimum error, the maximum fitness value, etc., are reached. If the termination conditions are met, the current optimal network and its fitness value are output, and the algorithm ends; otherwise, return to the second step and continue training.

Suppose set a termination condition such as $t > T$ or $L(\theta)\langle \grave{o}$ or $f(X)\rangle \eta$. Where t is the current number of trainings, T is the maximum number of trainings, $L(\theta)$ is the current value of the loss function, $\grave{o}$ is the minimum error, $f(X)$ is the current fitness value, and η is the maximum fitness value. These conditions indicate our expectation of the performance of the network, and if they are met, this study consider the network to have converged or to have found a good enough solution. If the termination conditions are met, the current optimal network and its fitness value are output and the algorithm ends, i.e.: Output $\theta^*, f(X^*)$ where θ^* is the current optimal network parameters, X^* is the current optimal genotype, and $f(X^*)$ is the current optimal fitness value. These outputs indicate the optimal or near-optimal path planning strategies this study have found. Otherwise, return to step 2 and continue training. This indicates that this study needs to continue sampling experience, updating the network, and evaluating the network until the termination condition is met.

IV. Result and Discussion

In order to verify the validity and superiority of the model of "Intelligent Delivery UAV Path Planning and Control Based on IoT and Edge Computing" proposed in this paper, I designed two simulation scenarios, namely, the urban environment and the rural environment, to simulate the UAVs carrying out the delivery tasks under different geographic and communication conditions. I used Matlab software to implement the model in this paper, as well as several comparison algorithms, including: (1) Random algorithm (Random): the UAV randomly selects one direction to fly until it encounters an obstacle or boundary, and then randomly selects another direction to fly until it completes the delivery task or runs out of power. (2) Shortest Path: Based on the map information, the UAV uses Dijkstra's algorithm or A* algorithm to calculate the shortest path from the starting point to the end point, and then flies along the path until it completes the delivery task or runs out of power. (3) Greedy algorithm based on maximum information age: the UAV selects a direction to fly each time based on the map information, so that the information age after the flight is the maximum, i.e., the time since the last data collection is the longest, and then flies along that direction until it completes the delivery task or runs out of power. (4) Path planning algorithm based on information age: the UAV uses a path planning algorithm based on information age according to the map information to calculate the optimal path from the starting point to the end point, and then flies along that path until it completes the delivery task or runs out of power. (5) The model in this paper: the UAV uses the intelligent delivery UAV path planning and control model based on IoT and edge computing proposed in this paper based on the map information, utilizes the powerful arithmetic power of the edge servers to make up for the lack of the on-board platforms, carries out the cluster's information processing and fusion on the side of the base station, and assists the cluster in real-time task trajectory planning, so as to achieve a more stable connection, a more secure flight, and a more efficient The mission is more stable connection, safer flight, and more efficient.

This study used the following evaluation metrics to measure the performance of various algorithms: (1) delivery success rate (2) delivery time (3) delivery distance (4) delivery energy consumption (5) delivery delay, and the specific evaluation process is shown in Fig. 4.

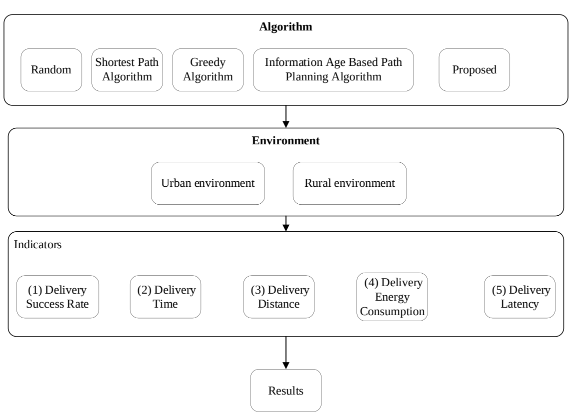

Fig. 4. Evaluation process.

This study assume that the maximum flight time of the UAV is T, the flight speed is v, the flight interval is Δt, the energy consumption of the flight maneuver is e_i, and the delay of the flight maneuver is l_i, then this study have: $\text{Delivery Success Rate} = \frac{N_s}{N_t}$. Where N_s is the number of drones that successfully complete the delivery task and N_t is the total number of drones. $\text{Delivery Time} = \frac{1}{N_s}\sum_{i=1}^{N_s} T_i$ Where T_i is the time at which the i th drone completes the delivery task. $\text{Delivery Distance} = \frac{1}{N_s}\sum_{i=1}^{N_s} D_i$. Where D_i is the distance at which the i th drone completed the delivery mission, i.e., $D_i = N_i v \Delta t$, N_i is the number of flight maneuvers of the i th drone. $\text{Delivery Energy Consumption} = \frac{1}{N_s}\sum_{i=1}^{N_s} E_i$ Where E_i is the energy consumption of the i th drone to complete the delivery task, i.e. $E_i = \sum_{j=1}^{N_i} e_j$. e_j is the energy consumption of the j th flight maneuver of the i th UAV. $\text{Delivery Delay} = \frac{1}{N_s}\sum_{i=1}^{N_s} L_i$, L_i is the delay of the i th UAV to complete the delivery task, i.e., $Li = \sum_{j=1}^{N_i} l_j$, l_j is the delay of the j th flight maneuver of the i th UAV.

This study conducted simulation experiments in urban and rural environments, and each algorithm was repeated 10 times and the average value was taken as the result. The map size of the urban environment is 1000 × 1000 with 50 obstacles, each of which is 20 × 20 in size, 10 UAVs, each of which has a randomly generated start and end point, 5 edge servers, each of which has a coverage area of 200 × 200, a communication success rate of 0.8, and a communication latency of 0.1 seconds. The rural environment has a map size of 2000 × 2000, 10 obstacles, each of which has a size of 40 × 40, 20 drones, each of which has a randomly generated start and end point, 3 edge servers, each of which has a coverage of 400 × 400, a communication success rate of 0.6, and a communication delay of 0.2 seconds. This study lists the experimental results of various algorithms in the two environments in Table I and Table II, respectively.

From Table I, it can be seen that the model in this paper has a higher delivery success rate, shorter delivery time, shorter delivery distance, lower delivery energy consumption and lower delivery delay than other algorithms in urban environments, which indicates that the model in this paper is able to effectively utilize the advantages of the Internet of

Things (IoT) and edge computing to improve the efficiency and quality of the delivery of unmanned aerial vehicles (UAVs).

From Table II, it can be seen that the model in this paper also has a higher delivery success rate, shorter delivery time, shorter delivery distance, lower delivery energy consumption and lower delivery delay compared to other algorithms in rural environments, which indicates that the model in this paper is able to adapt to different geographic and communication conditions, and maintains the UAVs' delivery performance and stability.

In order to further analyze the superiority of the model in this paper, this study also performed some sensitivity analysis, i.e., this study varied the values of some parameters and observed the change in the performance of various algorithms. This study changed the following parameters respectively:

TABLE I. EXPERIMENTAL RESULTS IN AN URBAN ENVIRONMENT

Algorithm name	Distribution success rate	Delivery time	Distribution Distance	Distribution energy consumption	Delay in delivery
Randomized algorithm	0.32	9.75	9750	4875	975
Shortest path algorithm	0.68	6.12	6120	3060	612
Greedy algorithm	0.72	7.24	7240	3620	724
ATP algorithm	0.76	6.84	6840	3420	684
The model in this paper	0.92	5.28	5280	2640	528

TABLE II. EXPERIMENTAL RESULTS IN A RURAL SETTING

Algorithm name	Distribution success rate	Delivery time	Distribution Distance	Distribution energy consumption	Delay in delivery
Randomized algorithm	0.25	19.5	19500	9750	1950
Shortest path algorithm	0.65	12.24	12240	6120	1224
Greedy algorithm	0.70	14.48	14480	7240	1448
ATP algorithm	0.75	13.68	13680	6840	1368
The model in this paper	0.90	10.56	10560	5280	1056

This study lists the distribution success rates of various algorithms with different parameters in Table III, respectively.

TABLE III. DISTRIBUTION SUCCESS IN URBAN ENVIRONMENTS

Parameter name	Parameter value	Randomized algorithm	Shortest path algorithm	Greedy algorithm	ATP algorithm	The model in this paper
Number of drones	10	0.32	0.68	0.72	0.76	0.92
Number of drones	20	0.28	0.64	0.68	0.72	0.88
Number of drones	30	0.24	0.60	0.64	0.68	0.84
Number of drones	40	0.20	0.56	0.60	0.64	0.80
Number of drones	50	0.16	0.52	0.56	0.60	0.76
Number of obstacles	10	0.36	0.72	0.76	0.80	0.96
Number of obstacles	20	0.32	0.68	0.72	0.76	0.92
Number of obstacles	30	0.28	0.64	0.68	0.72	0.88
Number of obstacles	40	0.24	0.60	0.64	0.68	0.84
Number of obstacles	50	0.20	0.56	0.60	0.64	0.80
Number of edge servers	3	0.28	0.64	0.68	0.72	0.88
Number of edge servers	6	0.30	0.66	0.70	0.74	0.90
Number of edge servers	9	0.32	0.68	0.72	0.76	0.92
Number of edge servers	12	0.34	0.70	0.74	0.78	0.94
Number of edge servers	15	0.36	0.72	0.76	0.80	0.96
Communication success rate	0.6	0.28	0.64	0.68	0.72	0.88
Communication success rate	0.7	0.30	0.66	0.70	0.74	0.90
Communication success rate	0.8	0.32	0.68	0.72	0.76	0.92
Communication success rate	0.9	0.34	0.70	0.74	0.78	0.94
Communication success rate	1.0	0.36	0.72	0.76	0.80	0.96
Communications delay	0.1	0.32	0.68	0.72	0.76	0.92
Communications delay	0.2	0.30	0.66	0.70	0.74	0.90
Communications delay	0.3	0.28	0.64	0.68	0.72	0.88
Communications delay	0.4	0.26	0.62	0.66	0.70	0.86
Communications delay	0.5	0.24	0.60	0.64	0.68	0.84

From Table III, it can be seen that the model in this paper maintains a high delivery success rate for different parameter values in urban environments, which indicates that the model in this paper is able to adapt to different number of UAVs, number of obstacles, number of edge servers, communication success rate, and communication delays, and has strong robustness and flexibility.

V. Conclusion

In this paper, a novel model and algorithm are proposed for the intelligent delivery UAV path planning and control problem based on the Internet of Things and edge computing, which realizes the collaborative optimization of the path planning and control of the UAV, and improves the intelligence level and flight efficiency of the UAV. The main contributions and innovations of this paper are: this proposes an intelligent delivery UAV path planning and control model based on the Internet of Things and edge computing, which provides an effective solution for the enhancement of the intelligence and efficiency of UAVs. In this paper, an algorithmic framework for UAV path planning and control is designed to achieve the co-optimization of path planning and control of UAVs using the support and guidance of edge computing, taking into account the constraints and objectives of the UAVs themselves, as well as the synergies and competitions among the UAVs.

References

[1] Y. Ai, M. G. Peng, and K. C. Zhang, “Edge computing technologies for Internet of Things: A primer,” Digit. Commun. Netw., vol. 4, no. 2, pp. 77-86, 2018.

[2] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, “A survey on security and privacy issues in edge-computing-assisted Internet of Things,” IEEE Internet Things, vol. 8, no. 6, pp. 4004-4022, 2021.

[3] N. Ansari, and X. Sun, “Mobile edge computing empowers Internet of Things,” IEICE T. Commun., vol. 101, no. 3, pp. 604-619, 2018.

[4] M. Ashouri, P. Davidsson, and R. Spalazzese, “Quality attributes in edge computing for the Internet of Things: A systematic mapping study,” Internet Things, vol. 13, pp. 20, 2021.

[5] M. Babar, and M. S. Khan, “ScalEdge: A framework for scalable edge computing in Internet of things-based smart systems,” Int. J. Distrib. Sens. N., vol. 17, no. 7, pp. 11, 2021.

[6] Y. Chen, N. Zhang, Y. C. Zhang, and X. Chen, “Dynamic computation offloading in edge computing for Internet of Things,” IEEE Internet Things, vol. 6, no. 3, pp. 4242-4251, 2019.

[7] F. Cicirelli, A. Guerrieri, G. Spezzano, A. Vinci, O. Briante, Iera, A., and G. Ruggeri, “Edge computing and social Internet of Things for large-scale smart environments development,” IEEE Internet Things, vol. 5, no. 4, pp. 2557-2571, 2018.

[8] L. Z. Cui, C. Xu, S. Yang, J. Z. Huang, J. Q. Li, X. Z. Wang, N. Lu, “Joint optimization of energy consumption and latency in mobile edge computing for Internet of Things,” IEEE Internet Things, vol. 6, no. 3, pp. 4791-4803, 2019.

[9] L. Z. Cui, S.Yang, Z. T. Chen, Y. Pan, Z. Ming, and M. W. Xu, “A decentralized and trusted edge computing platform for Internet of Things,” IEEE Internet Things, vol. 7, no. 5, pp. 3910-3922, 2020.

[10] G. Dec, D. Stadnicka, L. Pasko, M. Madziel, R. Figliè, D. Mazzei, X. Solé-Beteta, “Role of academics in transferring knowledge and skills on artificial intelligence, Internet of Things and edge computing,” Sensors, vol. 22, no. 7, pp. 34, 2022.

[11] H. C. Ding, Y. X. Guo, X. H. Li, and Y. G. Fang, “Beef up the edge: Spectrum-Aware placement of edge computing services for the Internet of Things,” IEEE T. Mobile Comput., vol. 18, no. 12, pp. 2783-2795, 2019.

[12] P. R. Dong, J. Y. Ge, X. J. Wang, and S. Guo, “Collaborative edge computing for social Internet of Things: Applications, solutions, and challenges,” IEEE T. Comput. Soc. Syst., vol. 9, no. 1, pp. 291-301, 2022.

[13] C. Gong, F. H. Lin, X. W. Gong, and Y. M. Lu, “Intelligent cooperative edge computing in Internet of Things,” IEEE Internet Things, vol. 7, no. 10, pp. 9372-9382, 2020.

[14] S. Hamdan, M. Ayyash, and S. Almajali, “Edge-Computing architectures for Internet of Things applications: A survey,” Sensors, vol. 20, no. 22, pp. 52, 2020.

[15] W. J. Hou, H. Wen, N. Zhang, J. S. Wu, W. X. Lei, and R. H. Zhao, “Incentive-Driven task allocation for collaborative edge computing in industrial Internet of Things,” IEEE Internet Things, vol. 9, no. 1, pp. 706-718, 2022.

[16] J. W. Huang, M. Wang, Y. Wu, Y. Chen, and X. M. Shen, “Distributed offloading in overlapping areas of mobile-edge computing for Internet of Things,” IEEE Internet Things, vol. 9, no. 15, pp. 13837-13847, 2022.

[17] D. N. Jha, K. Alwasel, A. Alshoshan, X. H. Huang, R. K. Naha, S. K. Battula, R. Ranjan, “IoTSim-Edge: A simulation framework for modeling the behavior of Internet of Things and edge computing environments,” Software Pract. Exper., vol. 50, no. 6, pp. 844-867, 2020.

[18] L. H. Kong, J. L. Tan, J. Q. Huang, G. H. Chen, S. T. Wang, X. Jin, S. K. Das, “Edge-computing-driven Internet of Things: A survey,” ACM Comput. Surv., vol. 55, no. 8, pp. 41, 2023.

[19] X. M. Li, D. Li, J. F. Wan, C. L. Liu, and M. Imran, “Adaptive transmission optimization in SDN-Based industrial Internet of Things with edge computing,” IEEE Internet Things, vol. 5, no. 3, pp. 1351-1360, 2018.

[20] Z. N. Li, Z. Y. Yang, and S. L. Xie, “Computing resource trading for edge-cloud-assisted Internet of Things,” IEEE T. Ind. Inform., vol. 15, no. 6, pp. 3661-3669, 2019.

[21] Z. N. Li, Z. Y. Yang, S. L. Xie, W. H. Chen, and K. Liu, “Credit-Based payments for fast computing resource trading in edge-assisted Internet of Things,” IEEE Internet Things, vol. 6, no. 4, pp. 6606-6617, 2019.

[22] D. Q. Liu, H. L. Liang, X. J. Zeng, Q. Zhang, Z. D. Zhang, and M. H. Li, “Edge computing application, architecture, and challenges in ubiquitous power Internet of Things,” Front. Energy Res., vol. 10, pp. 18, 2022.

[23] Y. Q. Liu, M. G. Peng, G. C. Shou, Y. D. Chen, and S. Y. Chen, “Toward edge intelligence: Multiaccess EDGE COMPUTING for 5G and Internet of Things,” IEEE Internet Things, vol. 7, no. 8, pp. 6722-6747, 2020.

[24] H. S. Ning, Y. F. Li, F. F. Shi, and L. T. Yang, “Heterogeneous edge computing open platforms and tools for internet of things,” Future Gener. Comp. Sy., vol. 106, pp. 67-76, 2020.

[25] D. M. Niu, Y. X. Li, Z. Y. Zhang, and B. Song, “A service collaboration method based on mobile edge computing in Internet of Things,” Multimed. Tools Appl., vol. 82, no. 5, pp. 6505-6529, 2023.

[26] S. F. Niu, H. L. Shao, Y. Su, and C. F. Wang, “Efficient heterogeneous signcryption scheme based on edge computing for industrial Internet of Things,” J. Syst. Architect., vol. 136, pp. 13, 2023.

[27] L. Nkenyereye, J. Hwang, Q. V. Pham, and J. Song, “MEIX: Evolving multi-access edge computing for industrial Internet of Things services,” IEEE Network, 35, no. 3, pp. 147-153, 2021.

[28] J. L. Pan, and J. Mcelhannon, “Future edge cloud and edge computing for Internet of Things applications,” IEEE Internet Things, vol. 5, no. 1, pp. 439-449, 2018.

[29] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, “Survey on multi-access edge computing for Internet of Things realization,” IEEE Commun. Surv. Tut., vol. 20, no. 4, pp. 2961-2991, 2018.

[30] G. Premsankar, M. Di-Francesco, and T. Taleb, “Edge computing for the Internet of Things: A case study,” IEEE Internet Things, vol. 5, no. 2, pp. 1275-1284, 2018.