Joint Deployment and Trajectory Optimization in UAV-Assisted Vehicular Edge Computing Networks

Zhiwei Wu, Zilin Yang, Chao Yang, Jixu Lin, Yi Liu, Xin Chen

***Abstract*—As the general mobile edge computing (MEC) scheme cannot adequately handle the emergency communication requirements in vehicular networks, unmanned aerial vehicle (UAV)-assisted vehicular edge computing networks (VECNs) are envisioned as the reliable and cost-efficient paradigm for the mobility and flexibility of UAVs. UAVs can perform as the temporary base stations to provide edge services for road vehicles with heavy traffic. However, it takes a long time and huge energy consumption for the UAV to fly from the stay charging station to the mission areas disorderly. In this paper, we design a pre-dispatch UAV-assisted VECNs system to cope with the demand of vehicles in multiple traffic jams. We propose an optimal UAV flight trajectory algorithm based on the traffic situation awareness. The cloud computing center (CCC) server predicts the real-time traffic conditions, and assigns UAVs to different mission areas periodically. Then, a flight trajectory optimization problem is formulated to minimize the cost of UAVs, while both the UAV flying and turning energy costs are mainly considered. In addition, we propose a deep reinforcement learning(DRL)-based energy efficiency autonomous deployment strategy, to obtain the optimal hovering position of UAV at each assigned mission area. Simulation results demonstrate that our proposed method can obtain an optimal flight path and deployment of UAV with lower energy consumption.**

***Index Terms*—Deep reinforcement learning, energy efficiency, mobile edge computing, unmanned aerial vehicle relay.**

I. INTRODUCTION

WITH the rapid development of information technology, the unprecedented popularity of smart mobile devices provides a powerful platform for new applications and also brings many novel challenges [1], [2]. As a typical Internet of things (IoT), Internet of vehicles (IoV) can realize ubiquitous information exchanging and content sharing between vehicles with almost no human intervention through its installed sensors and other smart devices [3]. With the help of LTE-V2X and the 5G NR-V2X technologies [4], the on-board units (OBU) installed on the vehicles connect with the roadside units (RSU), IoV can provide an informative travel environment for both drivers and passengers [5]. However, compared with cloud computing center (CCC) servers, OBUs usually have lower computing capacity [6], [7]. The main shortcoming of the direct connection between the vehicle and the CCC servers is the long distance communication delay. Due to the backhaul load during peak hours, it is difficult for CCC servers to meet the needs of various delay-sensitive mobile programs [8]. The tension between resource constrained vehicle-mounted terminals and computing intensive applications has become a bottleneck for improving user satisfaction and service quality in IoV. Vehicular edge computing networks (VECNs) are considered as the promising paradigm to solve the above challenges via deploying servers at the edge of the wireless access network, for example the RSUs [9]. In VECNs, the smart devices and users in vehicles can offload computing tasks to the nearby vehicles or the edge servers deployed at the RSUs, both the communication energy consumption and the delay between edge servers and users become lower.

Normally, the moving vehicles on the road, the parked vehicles at the roadside parking lots, and the RSUs can perform as edge nodes to provide communication and computation resources for the vehicles in VECNs [10]–[12]. However, the dynamic topology change of vehicular network makes the effective communication time duration of both vehicle to vehicle (V2V) and vehicle to roadside unit (V2R) extremely short. Moreover, the locations of the parked vehicles and RSUs are usually fixed, the deployment of MEC servers requires a certain amount of space and cost, which limits the capabilities of the edge servers [13].

In recent years, unmanned aerial vehicles (UAVs) have received extensive attention from academia and industry due to their strong mobility and flexible deployment. In [14], the author designed a data collection scheme based on blockchain, that uses UAVs to obtain data from IoT devices. It helps to improve the security of the IoT and the energy consumption is reduced. The author also extended the above scheme to the UAV swarm system in subsequent research [15]. In [16], the author proposed an AI-authorized pandemic monitoring program based on blockchain.

In cooperation with the CCC center, UAVs can operate as a moving edge server, the vehicles on ground can offload part of the computing tasks to the UAVs. In addition, the vehicles can use UAVs as relays, the tasks will be offloaded to other servers (such as cloud computing centers) to complete. For the situation that the UAV is far away from the mission vehicles, a huge energy and time costs exist when the UAV flies to the mission areas directly.

Manuscript received April 4, 2021; revised July 13, 2021; approved for publication by Ji-Woong Choi, Division II Editor, August 9, 2021.

This work was supported in part by the National Natural Science Foundation of China under Grants 62003094, 62003099, 61773126, and U191140003, in part by the Guangdong Basic and Applied Basic Research Foundation under Grants 2019A1515011377, and 2019A1515011114, in part by the Guangzhou Science and Technology Plan Project under Grant 202102020573, in part by the Key Research and Development Program of Guangdong Province under Grant 2019B010141001.

Z. Wu, Z. Yang, C. Yang, J. Lin, Y. Liu, and X. Chen are with the Guangdong Key Laboratory of IoT Information Technology, School of Automation, Guangdong University of Technology, Guangzhou. Email: wuzhiwei_wzw@outlook.com, yangzilin110@163.com, yangchaoscut@aliyun.com.

C. Yang is the corresponding author.

Digital Object Identifier 10.23919/JCN.2021.000026

1229-2370/21/$10.00 © 2021 KICS

In this paper, we propose a pre-dispatch UAV-assisted VECN system in an urban traffic environment. The UAVs can be assigned to different mission areas with traffic jams in advance. In detail, an optimal UAV flight trajectory algorithm is proposed based on the traffic situation awareness firstly. We formulate an optimization problem to obtain the optimal UAV flight trajectory, while both the flying and turning energy consumptions of UAVs are mainly considered. Genetic algorithm (GA) is used to find the problem solution. Then, for the UAV flight trajectory is obtained based on the predicted values, when the UAV arrives the mission area, it should find the optimal hovering position. We propose an autonomous deployment and energy efficiency optimization strategy based on deep reinforcement learning (DRL), the real time differentiated computation requests are considered. The main contributions are summarized as follows:

- We propose a pre-dispatch UAV-assisted VECN system. The CCC center predicts the short-time traffic conditions periodically, and the UAV can fly to the assigned mission areas in advance.
- We propose an optimal UAV flight trajectory algorithm based on the traffic situation awareness, while both the flying and turning energy consumption of the UAV are considered.
- We propose a real time control strategy to find the optimal hovering position of UAV in each mission area, a DQN-based hovering algorithm is introduced for the practical computation requests of road vehicles are considered.

The rest of this article is organized as follows. Section II summarizes recent research on UAV-enabled MEC networks. Section III introduces the system model including the UAV to road vehicles'channel and task transmission and computation energy consumption models. In Section IV, the problem formulation of this work is proposed. We give the details of the UAV flight trajectory algorithm and the optimal hovering position algorithm. In Section V, simulation results are presented to evaluate the proposed algorithms. Finally, we conclude the paper in Section VI.

II. Related Work

In recent years, UAVs have received extensive attention in wireless communication networks [17]–[19]. For the mobility and deployment flexibility, UAVs can perform as the temporary base stations (BSs) or relay nodes in areas with limited communication, such as emergency rescue after natural disasters and stadiums during sport events [20], [21]. Installed MEC servers on UAV, the UAV-assisted MEC networks have two advantages: 1) Compared with traditional ground base stations, UAVs can dynamically adjust their hovering positions in the air to provide better services, according to the actual environment and mission requirements. 2) It can establish better line-of-sight (LOS) communication with mobile users, and further shorten the transmission distance. In [22], the author analyzed the energy consumption of UAV as a relay node, the system energy efficiency was maximized via the joint transmission power of UAV and BS, the trajectory, acceleration and flight speed of UAV were considered. In [23], the author studied the joint design of computational offloading and resource allocation, as well as the UAV trajectory optimization, to minimize the UAV energy consumption and task completion time. In [24], the UAV performed computing tasks that the tasks had offloaded from the mobile terminal users (TU). The movement of each TU follows the Gauss-Markov random model. A QoS-based action selection strategy was developed based on the dual deep Q network (DQN) algorithm to maximize system rewards.

Thanks to the development of UAV flying and battery storage technologies [25]–[29], modern UAVs are capable of providing powerful computing capacity in complex environments [30]–[32]. Some research for the UAV-assist VECNs had been proposed [33], [34]. In [33], the authors proposed a software-defined networking(SDN)-enabled UAV-assisted vehicular computation offloading optimization framework to minimize the system cost of vehicle computing tasks. In [34], the authors proposed an edge computing architecture for UAV cluster service vehicles, an efficient multi-mode and multi-task offloading scheme is achieved.

However, the aforementioned studies only considered the energy consumption minimization, the communication and computation resource allocation schemes of the UAV after it reaches the mission area. A big assumption is that the UAVs can provide edge services in the vicinity when the computing requests of users or vehicles exist. For the situation that the UAV is far away from the mission area, the users with tasks need to wait for the dispatch center (i.e. CCC servers) to assign the UAV to task requesting areas. Undoubtedly, this will significantly affect the service quality of UAV, and it is not enough to reflect the superiority of UAV flexible scheduling. Specially, when there are multiple computing tasks in more than one mission area requests for an UAV at the same time, the UAV flight scheduling will become more complex. In addition, in order to enable the UAV to access all target mission areas within the minimum flight energy consumption, most of the existing related works treat the UAV path planning process as an approximate traveling salesman problem (TSP) [35], the energy consumption of the UAV flying between two nodes is usually replaced by the Euclidean distance. However, in practical applications, the flight distance is not the only factor that affects the flight energy consumption of UAV. The flight speed, hovering time and flight angle of the UAV will all affect the flight energy consumption. Especially on the crowded urban roads, the UAVs need to fly along the traffic roads, and the flight turning energy consumption needs to considered mainly. [36] considered the flight turning factors when planning the UAV flying path, however, the proposed algorithm just reduces the number of turns simply, the in-depth discussion of energy consumption influences caused by the flight turning hasn't been discussed.

Above all, few works considered the fly turning energy consumption in UAV path planning, and both the energy and time consumptions of UAV traveling from the control center to the mission locations. Compared with the above research, this paper have two innovations: 1) We propose a pre-scheduling UAV-assisted VECN system, which can schedule UAVs to mission locations in advance, reducing user's waiting time;

TABLE I
SUMMARY OF NOTATIONS AND SYMBOLS.

Notation	Representation of the symbol or symbol
$d_{n,n+1}$	UAV flying distance
e_v	Energy consumption coefficient of UAV flying straight at speed v
$E^{v}_{n,n+1}$	UAV straight flight energy consumption parameters
E^{turn}	Turning energy consumption of an UAV
$\theta_{g,g+1,g+2}$	The size of the turning angle
η_1	Coefficient of turning energy consumption
η_2	Coefficient of turning energy consumption
E_n^{serve}	UAV service energy consumption
E_n^{comm}	UAV communication energy consumption
E_n^{comp}	UAV computing energy consumption
N	Index of intersection (or mission location)
$\mathcal{M}_n$	The set of vehicles at mission area
h	The height of the UAV flight
P_{LOS}	The probability of LOS between UAV and vehicle
L_{LOS}	Path loss of LOS between UAV and vehicle
P_{NLOS}	The probability of NLOS between UAV and vehicle
L_{NLOS}	Path loss of NLOS between UAV and vehicle
$L(h, r_m)$	Total path loss between UAV and vehicle
d_m	The straight-line distance between the drone and the vehicle
P_t	UAV transmission power
P_r^m	The received power of the vehicle
$\mathcal{U}_n$	Distribution of ground vehicles
s_t	State space of DQN model
a_t	Action space of DQN model
r_t	Reward function of DQN model

TABLE II
TRAFFIC FLOW DATA.

Timestamp	CrossroadID	VehicleID	CrossroadLng	CrossroadLat
2019/8/1 13:28	100120	LU-U-3d1d4c9c9bc6a990	120.346987	36.090423
2019/8/1 16:56	100120	GUI-B-0ef8d356d7a038cb	120.346987	36.090423
2019/8/1 13:18	100120	JI-A-39803195c4a8785d	120.346987	36.090423
2019/8/1 13:31	100120	LU-B-d79f5f83d2133418	120.346987	36.090423
2019/8/1 11:38	100359	LU-B-1be0e79deaaeb7e2	120.409793	36.102926
2019/8/1 12:49	100359	LU-U-195fdfdc3f50b37c	120.409793	36.102926
2019/8/1 08:25	100359	LU-B-37f3d431a4f632a6	120.409793	36.102926
2019/8/1 15:46	100359	LU-B-61f561d4296cecd3	120.409793	36.102926
2019/8/1 16:48	100359	LU-B-45a27b22eb8902e6	120.409793	36.102926
2019/8/1 16:01	100349	LU-B-0e14e01825e34446	120.426224	36.173504
2019/8/1 08:40	100349	LU-B-52220194a04d0349	120.426224	36.173504
2019/8/1 18:45	100349	LU-B-b5cbdd623826f1aa	120.426224	36.173504
2019/8/1 13:07	100349	LU-B-4721537f2f0776a8	120.426224	36.173504
2019/8/1 13:17	100349	LU-B-4e0919395fa6ddf7	120.426224	36.173504

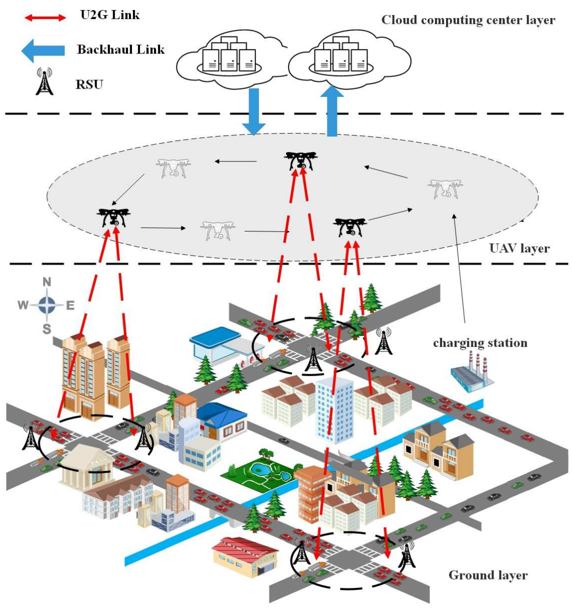

Fig. 1. System model.

2) We plan the UAV's path under the premise of considering the fly turning energy consumption, which reduces the UAV's non-service energy consumption.

III. System Model

A. Framework of System

As shown in Fig. 1, we consider an UAV-assisted VECN system in the urban traffic environment. It has a three-layer network architecture, including the CCC server layer, UAV layer, and ground layer with multiple vehicles on the road. The CCC server acts as the central controller to coordinate the operation of the entire system. UAVs will be charged at the charging station and wait for the next mission instruction. A large number of sensing devices and moving vehicles on the road and the RSUs constitute the underlying equipment of the intelligent transportation system (ITS), the traffic information is collected in the cloud server through the air-to-ground (A2G) and backhaul links. We consider that the vehicles moving on the road generate kinds of computing tasks continuously [37]. During off-peak hours, when there are fewer vehicles on the road, the RSUs can provide sufficient computing and communication services for the covered vehicles. However, when the traffic jams occurs on the road, especially in the road intersections, more vehicles computing task requests exist. Combined with the RSU, the UAV can perform as the temporary BS and provide timely computation and communication services for the ground vehicles in different mission areas, through the arrangement and guidance of the CCC center.

The UAV-assisted VECN system operates in a slot-by-slot fashion with fixed length time durations. At the beginning of each slot, the cloud center predicts the traffic conditions of the covered road intersections, via performing data analysis and processing of the collected historical traffic flow data. Historical data is collected through ITS (such as monitoring equipment at intersections). In our work, we use historical traffic data within 7 days to predict the traffic volume of each intersection every 30 minutes in the next day. The data set comes from the competition website DCLab [43], and the specific data is shown in the Table II (the vehicle ID has been coded to avoid leakage of owner information).

Then, the objective road intersections that may with heavy traffic are obtained. The cloud center schedules UAVs to different mission areas based on the flying distance and the battery capacity. The optimal UAVs assignment scheme will not be discussed in this paper. We focus on the scenario that an UAV is scheduled to provide edge services for a set of road intersections. According to the predicted traffic conditions, we obtain the estimated computing task requests in each mission area and an optimal UAV flying trajectory from the staying station and the assigned road intersections is calculated. When a UAV arrives at a new mission area, supported by the ITS, the UAV can obtain the traffic information, such as the number of ground vehicles, and fly to the optimal position to hover to provide communication and computation services for ground vehicles. After completing the task of the current intersection, the UAV will fly to the next task mission area according to the proposed path plan. When the tasks of all road intersections are completed, the UAV flies back to the staying station and

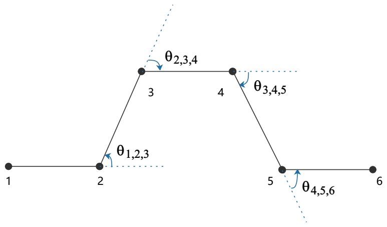

Fig. 2. The specific definition of $\theta_{g,g+1,g+2}$.

waits for the next task assignment.

B. Energy Model of UAV

In this work, we consider the UAV flying trajectory and hovering position optimization problem, the energy consumption becomes a key factor affecting the QoS of UAV, in which the flight energy consumption accounts for the highest proportion. In order to analyze the flight energy consumption of UAV, it can be further divided into horizontal flight, vertical flight and turning energy consumption. In the existing UAV-assisted MEC system, the flight energy optimization of the UAV only considers the horizontal flight and the vertical flight energy consumption, and does not take into consideration the turning energy consumption. Actually, when the UAV flies over a corner, the additional turning energy consumption occurs. In the urban road environment, the UAV flies between the mission areas along the urban roads. Since frequency lifting and lowering operations cause a lot of energy loss, it is reasonable and necessary to consider the turning energy consumption mainly.

In our work, we set that the UAV is flying at a constant speed at a fixed height h. The energy consumption of the UAV flying in a straight line at a fixed speed is denoted as

$$E^{v}_{n,n+1}(d) = e_v d_{n,n+1}, \tag{1}$$

where $d_{n,n+1}$ is the distance of the UAV flying from mission area n to area $n+1$, $n \in \{1, 2, \cdots, N\}$. and e_v is the unit energy consumption per meter when UAV is flying in a straight line at a constant speed v m/s. In addition, the turning energy consumption is denoted as [38]

$$E^{\text{turn}}_{g,g+1,g+2}(\theta) = \eta_1 \theta^2_{g,g+1,g+2} + \eta_2 \theta_{g,g+1,g+2}, \tag{2}$$

where $\theta_{g,g+1,g+2}$ is the angle of the UAV when it turns, the specific definition is shown as the Fig. 2. η_1 and η_2 are the fitting parameters obtained from experiments [38].

Hovering energy consumption is related to the computing task requirements in the mission area. Since we consider a pre-dispatch UAV-assisted VECN system, the edge service energy consumption of the mission areas are estimated in advance to facilitate the optimal route planning. In the forecasting stage, we set that the task demands of vehicles in a traffic intersection is related to its historical communication and calculation demands. Set E^{server}_n as the energy consumption when the UAV is hovering at the mission area n and provides edge services to the road vehicles. E^{comm}_n and E^{comp}_n denote the energy consumptions of processing the historical communication demand and the historical computation demand, E^{hover}_n is the hovering energy consumption, we have:

$$E^{\text{server}}_n = E^{\text{comm}}_n + E^{\text{comp}}_n + E^{\text{hover}}_n. \tag{3}$$

C. Channel Model of UAV

After the UAV arrives the mission area, it provides edge services for the covered vehicles on the road. The communication channel model between the UAV and the ground vehicles is considered. We define $\mathcal{M}_n$ as the set of vehicles at mission area n, $\mathcal{M}_n = \{1, 2, \cdots, m, \cdots, M_n\}$. Since the hovering UAV is regarded as a temporary aerial BS, according to the A2G model in [39], the probability of the LOS link between the m-th vehicle and the UAV is denoted as

$$P_{LOS}(r_m, h) = \frac{1}{1 + \alpha \exp\left\{-\beta\left(\arctan\left(\frac{h}{r_m}\right) - \alpha\right)\right\}}, \tag{4}$$

where α and β are constants related to the operation environment, h represents the height of the UAV, and r_m is the horizontal distance between the m-th vehicle and the UAV. The calculation of r_m is denoted as

$$r_m = \sqrt{(m_x - u_x)^2 + (m_y - u_y)^2}, \tag{5}$$

where (m_x, m_y) represents the position of the m-th vehicle on the horizontal, and (u_x, u_y) is the position of the UAV on the horizontal. In addition, the probability of non-line-of-sight (NLOS) communication link is

$$P_{NLOS} = 1 - P_{LOS}, \tag{6}$$

Taking into account the long-term changes of the channel and the average path loss, the path loss models of LOS and NLOS in the UAV [39] are denoted as

$$L_{LOS} = 20 \log\left(\frac{4\pi f_c d_m}{c}\right) + \eta_{LOS}, \tag{7}$$

$$L_{NLOS} = 20 \log\left(\frac{4\pi f_c d_m}{c}\right) + \eta_{NLOS}, \tag{8}$$

$$d_m = \sqrt{h^2 + r_m^2}, \tag{9}$$

where f_c is the transmission carrier frequency, d_m is the straight-line distance between the UAV and the m-th vehicle. Under both the LOS and NLOS models, the average path loss of A2G link can be denoted as

$$L(h, r_m) = L_{LOS} P_{LOS} + L_{NLOS} P_{NLOS}. \tag{10}$$

For a given UAV transmit power P_t, the received power of the m-th vehicle depends on the path loss experienced by its communication link [39], which is denoted as

$$P^m_r = P_t - L(h, r_m). \tag{11}$$

According to (4) and (10), we analyse the probability of LOS and path loss between UAV and vehicles in different environments (i.e., suburbs, cities, dense cities, and highly dense cities) with the elevation angle of the vehicle to the UAV, based on the public data sets [39], [40]. The simulation results

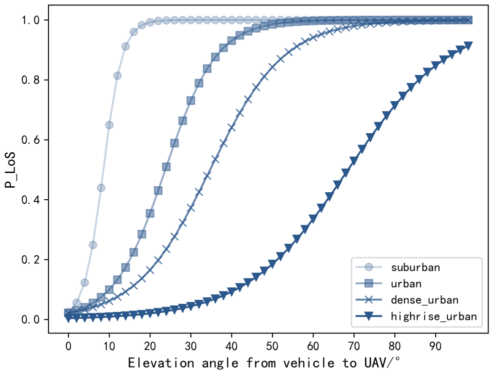

Fig. 3. Probability of LoS in different environments.

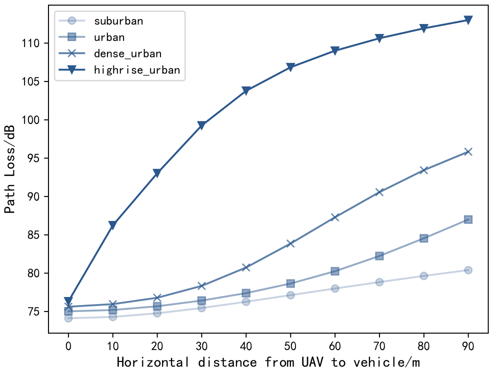

Fig. 4. Path loss in different environments.

are shown as Fig. 3 and Fig. 4. As the density of ground equipment increases, the LOS communication probability between vehicles and UAVs is decreasing gradually. However, when the UAV is flying directly above the vehicle, it can maintain LOS communication, which also reflects the advantage of UAV as a dynamic temporal edge node. On the other hand, as the elevation angle increases, the probability of line-of-sight communication between the UAV and vehicle also increases, and gradually converges to 1. For the communication path loss between the vehicle and UAV. When the horizontal distance between the UAV and vehicle increases, the path loss becomes higher. It is necessary to find the suitable and optimal hovering position of the UAV to provide edge computing servers.

IV. Problem Formulation and Solution

In the proposed UAV-assisted VECN system, under the guidance of the CCC servers, the UAVs can provide temporary edge services for vehicles on congestion road intersections. We consider a scenario that an UAV is assigned to a set of mission areas. An optimal UAV flight trajectory strategy is proposed based on the traffic situational awareness. When the UAV arrives the mission area, an optimal hovering position selection algorithm is proposed for the UAV. In detail, the system operation includes three main problems: The traffic flow forecasting in the CCC servers, the UAV flight trajectory optimization and the UAV hovering position optimization problems.

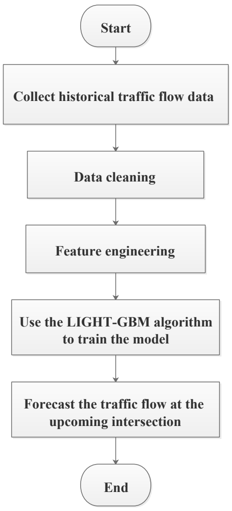
Start
Collect historical traffic flow data
Data cleaning
Feature engineering
Use the LIGHT-GBM algorithm
to train the model
Forecast the traffic flow at the
upcoming intersection
End

Fig. 5. Traffic flow prediction flow chart.

A. *Traffic Flow Forecasting*

The traffic flow prediction flow chart is shown as Fig. 5, the stpdf of traffic flow prediction are mainly divided into three aspects: Data cleaning, feature engineering, and model training and prediction.

Data cleaning is mainly to clean the raw data, such as duplicate data generated by data collection, abnormal data, etc. Feature engineering is mainly to aggregate data, calculate the traffic volume of each intersection every 30 minutes, and carry out relevant feature extraction and combination, such as the extraction of time information (year, month, day, whether it is a working day, etc). Then, we train the model and test the prediction results.

We use the light-GBM algorithm to predict traffic flow, which is an excellent algorithm proposed by Microsoft in 2017 to quickly process large-scale data [41]. The model utilizes the root mean square error (RMSE) as the evaluation standard. RMSE is the square root of the ratio of the square of the deviation between the predicted value and the true value to the number of observations. In actual measurement, the number of observations is always limited, and the true obtained value can only be replaced by a reliable value. The RMSE is highly sensitive to the huge large or small errors in a group of measurements, so it can well reflect the precision of the measurement. The calculation formula of RMSE is

$$RSME = \sqrt{\frac{\sum_{k=1}^{K}(z_{\text{pred}} - z_{\text{true}})^2}{K}}, \tag{12}$$

where z_{pred} represents the predicted value, z_{true} represents the true value, and m represents the number of observations.

After the prediction, we obtain the traffic flow of multiple different road intersections, including the number of vehicles passing through the intersection in a unit time. We consider that the computing task requests of vehicles are generated continuously, the whole task demands are related to the flow of vehicles. For a given threshold of the traffic volume, when the traffic volume at an intersection reaches this threshold, we consider the intersection is in the communication congestion state and mark it as a mission area.

B. UAV Flight Trajectory Optimization

When the CCC assigns a set of mission areas for the UAV, an optimal UAV flight trajectory algorithm is proposed to guide the UAV. Although the battery capacity has been improved, in order to enable UAVs to provide higher-quality computing or offloading services with less energy consumption, the energy consumption of UAVs is worth considering. In our work, we focus on the urban traffic environment, the UAV flies along the urban roads at a fixed altitude, and needs to turn multiple corners. Thus, in the designed algorithm, not only the regular straight flying energy consumption but also the turning energy consumption also be considered. The quality of service (QoS) function of UAV can be defined as

$$\eta = \frac{\sum_{n=1}^{N} E_n^{\text{serve}}}{E^Q + \sum_{n=1}^{N} E_n^{\text{serve}} + \sum_{n=1}^{N} E_n^{\text{hover}}}, \tag{13}$$

where Q is the UAV flight trajectory, E^Q represents the flight energy consumption of the UAV under trajectory Q, Since the estimated service demands are obtained through traffic flow prediction, the optimization problem becomes to maximize the QoS of UAV via optimizing the trajectory energy consumption E^Q, is described the following problem **P1**.

$$\max_{E^Q} . \, \eta,$$

subject to

$$E^Q + \sum_{n=1}^{N} E_n^{\text{serve}} \leq E, \tag{14}$$

where in (14), E is the battery storage of the UAV. It is necessary to ensure that the total service energy consumption and flight energy consumption are less than the energy stored by the battery E. And we have

$$E^Q = E^S + E^T, \tag{15}$$

$$E^S = E_{N,0}^v(d) + \sum_{n=1}^{N-1} E_{n,n+1}^v(d), \tag{16}$$

$$E^T = E_{N-1,N,0}^{\text{turn}} + \sum_{n=1}^{N-2} \sum_{j \in \mathcal{J}} E_{j,j+1,j+2}^{\text{turn}}, \tag{17}$$

where E^S denotes the energy consumption of UAV straight fight, that is associated with the flying distance. E^T denotes the UAV turning energy consumption. The formulations of $E_{n,n+1}^v(d)$ and $E_{j,j+1,j+2}^{\text{turn}}$ are shown as (9) and (10). $\mathcal{J}$ denotes the set of UAV flying pass corners. Moreover, $E_{N,0}^v(d)$ and $E_{N-1,N,0}^{\text{turn}}$ denote the energy consumption when the UAV completes its mission and returns to the staying station.

In the problem **P1**, the UAV traverses along the road intersections (including the mission road intersections) and then flies back to the staying station, it is a typical TSP problem. In order to solve this problem, we propose an optimal UAV Flight Trajectory based on the traditional Genetic Algorithm, named as "UAV FT-GA". The description of the UAV FT-GA structure is given in Algorithm 1.

Algorithm 1 UAV FT-GA

```
Input: Crossover probability $P_c$, Mutation probability $P_m$,
    Population size $S$, Number of iterations $G$.
 1: Initialize each parameter and coding;
 2: create the initial population pop;
 3: repeat
 4:   Calculate fitness function according (18);
 5:   repeat
 6:     Select ttwo individuals from the population pop according to the fitness ratio selection algorithm;
 7:     if random $(0,1) \leq P_c$ then
 8:       Perform crossover operation on two individuals;
 9:     end if
10:     if random $(0,1) \leq P_m$ then
11:       Perform mutation operation on two individuals;
12:     end if
13:     Add two new individuals to the population newpop;
14:   until $S$ children are created
15: until The number of iterations exceeds $G$
Output: The optimal UAV flight trajectory and corresponding
    energy consumption $E^Q$ .
```

The fitness function of the UAV FT-GA algorithm in this article is related to the energy consumption of the UAV route. In order to minimize the flight energy consumption, we design the following fitness function,

$$fitness = \frac{1}{E^Q + \sum_{n=1}^{N} E_n^{\text{serve}}}, \tag{18}$$

where E^Q is the straight-line flight energy consumption in the UAV flight path and the corner energy consumption, E_n^{hover} is the hovering energy consumption, E_n^{hover} is the service energy consumption, and n is the task location number.

In Algorithm 1, we define and initialize each parameter, such as crossover probability P_c, mutation probability P_m, population size S, number of iterations G. Then we create the initial population. Lines 3 to 15 are the main loop part of the genetic algorithm, including selection, crossover, and mutation operations. First, the fitness value of the individual needs to be calculated, then the individual with the higher fitness value is selected as the parent. Parents need to perform crossover and mutation operations to obtain individual offspring. Crossover is the partial exchange of the gene sequences of the two parents according to different crossover operators, and mutation is the replacement operation of partial values of the parent gene sequence. Repeat the above steps until the iteration stops, and output the UAV flying path with the least energy consumption.

C. UAV Hovering Position Optimization

When the UAV arrives at the n-th road intersection, $n \in \{1, 2, \cdots, N\}$, it needs to find the best hovering position according to the actual traffic volume and mission requirements. Inspired by the maze game on the OpenAI website,

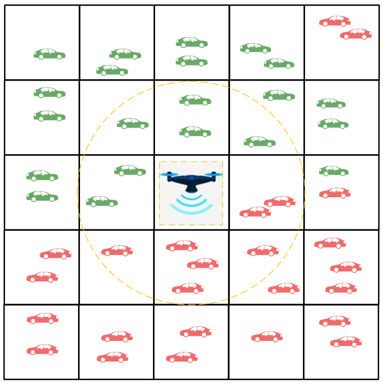

Fig. 6. The cover of UAV for a mission area.

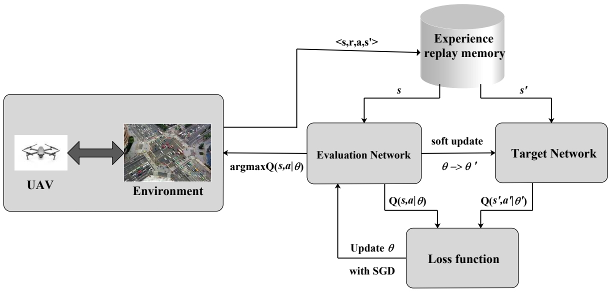

Fig. 7. DQN-based hovering algorithm framework.

we modeled the optimization problem of the UAV's optimal hovering position into an optimization model similar to the maze problem [42]. The maze game is a treasure hunting game. The explorer needs to continuously explore in an unknown maze until he finds the treasure. In this article, we assume that the UAV is a treasure hunter, and the best hovering position is the treasure that the UAV needs to find. The nth mission area is divided into $I \times J$ units, and the UAV is deployed above the unit and can cover the entire unit, as shown in the Fig. 6. Then, the best position deployment problem of the UAV can be modeled as a maze problem, and the existing maze solving method can be used.

Given the location of a ground vehicle, we use $\mathcal{U}_n = \{u_{11}, u_{12}, \cdots, u_{i,j}\}$ to represent the set of indicator variables for the distribution of ground vehicles. When there are k vehicles in the unit $\{i, j\}$, the indicator variable $u_{i,j} = k$, otherwise it is 0.

According to the above analysis and the proposed environment, we use the deep reinforcement learning method (i.e. DQN) to find the optimal hovering position of the UAV. The core idea is to use the Q function value network as the evaluation module. Based on the value network, we traverse various actions in the current observation state, and the environment interacts in real time. The state, action, reward and punishment values are stored in the playback memory unit, the Q-learning algorithm is used to repeatedly train the value network, and finally the action that can obtain the best value is selected to deploy the UAV. Fig. 7 shows the framework of DQN-based UAV optimal hovering algorithm. Among them, the DQN model is represented as a set.

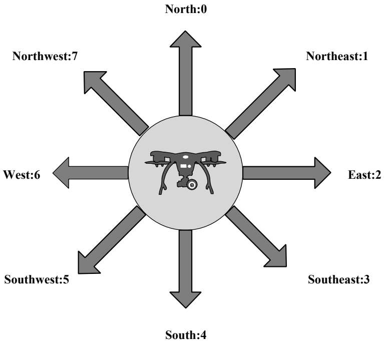

Fig. 8. UAV horizontal movement direction diagram.

Under the DQN framework, we define the following variables: State space, action space and reward function.

1) State Space: The status $\boldsymbol{s_t}$ consists of four parts, namely: location and distribution of ground vehicles $\mathcal{M}_n$,$\mathcal{U}_n$,the position of the UAV at the initial time (u_x^0, u_y^0, h) and the position at time t (u_x^t, u_y^t, h).

$$\boldsymbol{s_t} = \{\mathcal{M}_n, \mathcal{U}_n, (u_x^0, u_y^0, h), (u_x^t, u_y^t, h)\},$$

Through the defined state, the DRL agent can make decisions based on the current distribution of ground terminals, the location of the UAV, the calculation power and energy consumption.

2) Action Space: Corresponding to eight directions, we discretize the UAV movement and define eight horizontal movement directions, as shown in Fig. 8. Assuming that the UAV moves a fixed distance for each action, according to the feedback information of the environment, if the UAV does not reach the optimal position after performing the action, the UAV will continue to take corresponding actions until it arrives the optimal position and completes autonomy deployment. The overall action space is

$$a_t \in \{0, 1, 2, 3, 4, 5, 6, 7\}.$$

3) Reward Function: The reward function r_t is defined as

$$r_t = \frac{P_r^t}{s_p} - \frac{e_u^t}{s_e},$$

where P_r^t is the total received power of the vehicles covered by the UAV at time t, e_u^t denotes the flight energy consumption of the UAV from the starting point to the selected location at time t. s_p, s_e are used to standardize the vehicle received power and the UAV flight energy consumption related super parameter. When the reward r_t becomes larger, the vehicles on the road can get better communication services. At the same time, in order to prevent the UAV flying out of the mission area during the service process, a penalty value $-r_t$ is given and the current exploration is stopped.

The choice of action is related to the instant reward, but our goal is to obtain the largest future reward, so we define the Q function, the initial value of the Q function is 0, and the update process is as follows

$$\begin{aligned} Q_{new}(s_t, a_t) =& Q_{now}(s_t, a_t) + \lambda[r_t + \gamma \underset{a_{t+1}}{max}\, Q(s_{t+1}, a_{t+1}) \\ &- Q_{now}(s_t, a_t)], \end{aligned} \tag{19}$$

Algorithm 2 DQN-based hovering algorithm for autonomous deployment position of UAV

```
1: Initialize $Q$-network with weights $\mu$ and experience replay memory;
2: for time frames $t = 0, 1, \cdots, T$ do
3:     observe the initialize state $s_0$;
4:     Select a random action $a_t$ with probability $\epsilon$, otherwise choose action $a_t = \underset{a}{\operatorname{argmax}} Q(s_t, a, \mu_t)$;
5:     Perform action $a_t$, get reward $r_t$ and new state $s_{t+1}$;
6:     Store the experience $(s_t, a_t, r_t, s_{t+1})$ into the experience replay memory;
7:     Get a batch of samples from the replay memory, and calculate loss function $Loss(\mu_t)$ according (21);
8:     update $\mu_t$ according (23);
9: end for
```

where $Q_{now}(s_t, a_t)$ represents the Q value of current state, $Q(s_{t+1}, a_{t+1})$ represents the Q value of the next state, λ represents the learning rate, and γ represents the decay rate of the reward. The status update process is as follows

$$s_t \xrightarrow{a_t} r_t, s_{t+1}. \tag{20}$$

Due to the large state space and action space, directly defining the Q table to solve the problems is not effective. We use a neural network to approximate the Q function as $Q_(s_t, a_t) \approx Q_(s_t, a_t, \mu)$. Among them, μ is a network parameter. To minimize the gap between the two values, a loss function is defined for optimization, as

$$Loss(\mu_t) = E[(y_t - Q(s, a; \mu_t))^2], \tag{21}$$

where y_t is the Q value of the target network, and the calculation method is as follows:

$$y_t = r_t + \gamma \underset{a_{t+1}}{max} Q(s_{t+1}, a_{t+1}; \mu_t^-). \tag{22}$$

After that, we use the gradient descent method to update the μ as follows:

$$\mu_{t+1} = \mu_t + \lambda[y_t - Q(s, a; \mu_t)]\nabla_{\mu_t} Q(s, a; \mu_t). \tag{23}$$

DQN uses two neural networks: the target network and the main network. The target network is used to generate y_t, which is the evaluation benchmark for the loss function of the main network. At regular intervals, the parameters of the main network are assigned to the target network. After introducing the target network, in a certain period of training, the target Q value is kept unchanged, which reduces the correlation between the current and the target Q value to a certain extent, and improves the stability of the algorithm.

In order to avoid reaching the local optimal point, we adopt the $\epsilon -$ greedy policy, which is to choose random actions with probability ϵ. Moreover, in order to speed up the DQN algorithm convergence, a step penalty mechanism is also introduced. The DQN also introduces an experience replay memory, which stores the data obtained from the system exploration environment, the problems of correlation and non-static distribution are solved. It stores the transferred samples obtained from the interaction between the agent and the environment at each time step and stores them in the playback memory network. For a part of the data (minibatch) is randomly taken out with training, the correlation in the samples is disrupted and the stability of the algorithm is improved.

TABLE III
SIMULATION PARAMETERS.

Parameter	Defifinition	Value
h	The flying height of the UAV	50 m
d_{unit}	Cell side length	20 m
P_t	UAV transmission power	1 W
f_c	UAV launch frequency	2.4 GHz
η_{LOS}	Additional path loss under LOS	1.6
η_{NLOS}	Additional path loss under NLOS	23
α	Environmental parameters	12.08
β	Environmental parameters	0.11
s_p	Power standardized parameters	1.26×10^{-7}
s_e	enenry standardized parameters	0.33
η_1	UAV corner energy consumption parameters	4.426×10^{-6}
η_2	UAV corner energy consumption parameters	1.7738×10^{-4}
e_v	UAV straight flight energy consumption parameters	0.006

The description of the DQN-based hovering algorithm structure is given in Algorithm 2. Lines 2 to 9 are the iterative process of DQN. From line 4, UAV explores the environment according to the greedy strategy. When the random number is less than the greedy rate, it chooses a direction to explore randomly. Otherwise, it chooses the action $a_t = \underset{a}{\operatorname{argmax}} Q(s_t, a, \mu_t)$. Line 6 is to store the explored data in the experience replay memory. Lines 7 to 8 are to update network parameters. According to the proposed DQN algorithm, the UAV can find the optimal hovering position automatically.

V. NUMERICAL RESULT

In this section, we validate the effectiveness and convergence of the proposed algorithms through numerical simulation. We use python3.7 and TensorFlow2.0 to build a DQL environment, the experimental running system is window10, the CPU used is AMD R5-3600, and the GPU is NVIDA GTX1660Super. We select the real traffic data set from the competition public dataset [43] for simulation experiments. Other relevant parameter settings are shown in Table III.

A. Traffic Flow Forecasting

Fig. 9 is the visualization of the LightGBM training process. In the figure, the horizontal axis is the number of iterations, and the vertical axis is the evaluation function (i.e., RSME). The red curve named *valid_score* is the RMSE change of the *valid* set. We can find that the LightGBM algorithm has a good training effect and convergence performance. After about 350 iterations, the proposed LightGBM becomes convergence, the values of RMSE stabilize at around 2.3. Therefore, the LightGBM can be used to perform the traffic flow forecasting effectively, and the predictions of the traffic computing task requests in road intersections are obtained.

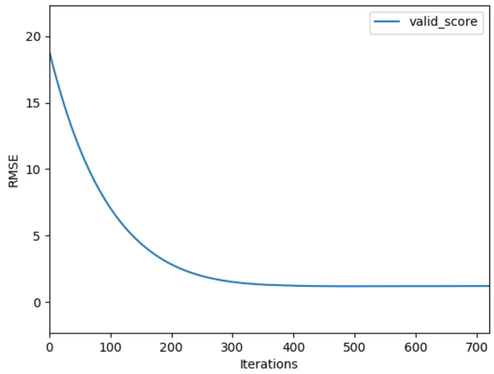

Fig. 9. LightGBM training process.

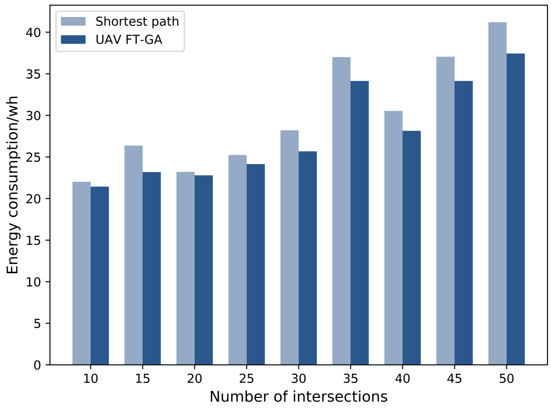

Fig. 10. Comparison of energy consumption for the proposed UAV FT-GA scheme and the shortness path scheme.

B. UAV FT-GA

Fig. 10 is a comparison diagram of actual energy consumption of the two schemes to obtain the best path by considering the turning energy consumption or not. In the latter comparison scheme, the UAV flies with the shortest path, and the turning energy consumption is not considered. In this simulation example, the coordinate points are randomly generated on the 1000×1000 terrain. GA is used to solve the problem **P1**, and the optimal flight path is obtained. In Fig. 10, the abscissa is the number of road intersections. From the figure, we can find that compared with the scheme without considering the turning energy consumption, the path planning of the UAV can significantly reduce the flight energy consumption in the proposed UAV GA-FT algorithm. The reason is that we consider the UAV flies along the urban roads, and much more corners should be passed. Considering the turning energy consumption, the proposed flight path planning scheme can be consistent with the actual traffic road situations. Fig. 11 is a schematic diagram of the GA iterative process considering the turning energy consumption. As shown in the figure, when the number of iterations reaches about 600, the proposed UAV FT-GA scheme becomes converge, indicating that the scheme has a good convergence performance.

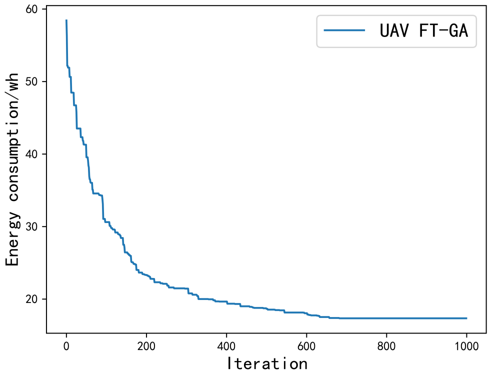

Fig. 11. Iteration of FT-GA.

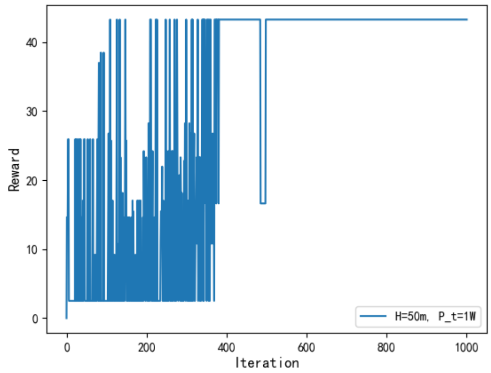

Fig. 12. DQN iterative process.

C. DQN-based Hovering Algorithm

When the UAV arrives the mission area, a DQN algorithm for autonomous deployment position of UAV is proposed to find the optimal hovering position of UAV. In order to validate the effectiveness of the proposed DQN-based hovering algorithm, we compare it with other existing algorithms:

- **Theoretical maximum:** In the simulation experiment, we know the environmental data in advance, so we can obtain the theoretical best hovering position in the current map and the theoretical maximum reward value corresponding to the location through detailed calculations.
- **Center layout method:** In this method, the UAV is deployed at the center of the map.
- **Q-Learning**: Q-Learning is a value-based learning algorithm in RL, which is used to find the optimal hovering position of UAV.
- **Sarsa**: SARSA (State-Action-Reward-State-Action) is an algorithm for learning Markov decision process strategies, which is used to find the optimal hovering position of UAV.

Fig. 12 is a simulation diagram of proposed DQN-based hovering algorithm. From the simulation results, we can find that the UAV reward value presents an oscillating changing at the initial moment. When the number of iterations reaches

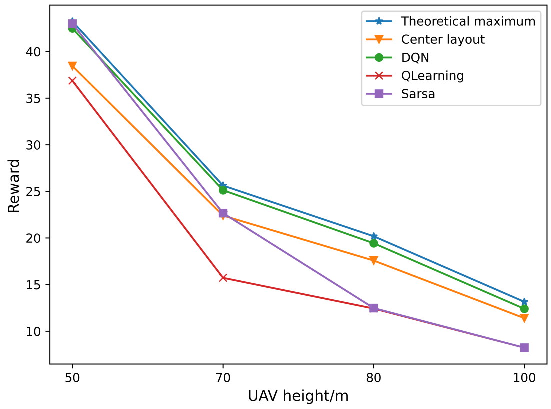

Fig. 13. Rewards of different height schemes.

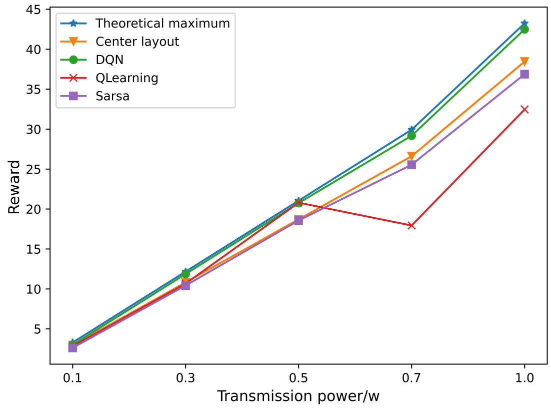

Fig. 14. Rewards of different transmission power schemes.

TABLE IV
ALGORITHM RUNNING TIME.

Algorithm	Iterations	Running time(s)
UAV FT-GA	1000	71.2595
DQN	1000	18.8437

around 600, the reward value tends to stabilize. This is because the UAV needs to constantly explore the surrounding environment in the early stage. When the enough corresponding data is collected, the UAV can find the optimal hovering position, complete its autonomous deployment, and provide services for the vehicles on the road.

Before the comparison experiment, we used the parameters in Table III to simulate, and the running time of the simulation program is shown in Table IV. Fig. 13 is the comparison of the reward value of our proposed DQN algorithm and other algorithms under different UAV flight heights. We can find that the proposed DQN algorithm can always obtain higher reward values than others, and be close to the theoretical maximum reward value. Q-Learning and Sarsa algorithms can also obtain the same value as DQN, but they are not stable enough compared with DQN.

Fig. 14 is the comparison of the reward value of our proposed DQN algorithm and other algorithms at different UAV transmission powers. Similar to Fig. 13, we can find that the DQN algorithm we proposed has always been able to obtain higher reward values than others , and be close to the theoretical maximum reward value. The main reason is that the proposed DQN algorithm uses two neural network fitting functions. The experience pools are used to store the explored data is also an important reason for improving stability and better rewards compared to other algorithms.

VI. CONCLUSION

Taking into account the impact of the UAV flying distance between the staying station to the mission areas, we propose a pre-scheduled UAV-assisted VECN system in urban traffic environment. The CCC uses the powerful data processing capabilities to predict traffic conditions of the covered road intersections, and assign UAVs to different predicted mission areas in advance. An optimal UAV flight trajectory strategy is proposed based on the traffic situational awareness. We design an UAV FT-GA algorithm to find the optimal flight trajectory, while the turning and flying energy consumption is considered. Then, when the UAV arrives the mission area, an optimal DQN-based UAV hovering position selection algorithm is proposed. Simulation results show that our proposed method can help the UAV obtain better service quality.

REFERENCES

[1] V. R. Warriar, J. R. Woodward, and L. Tokarchuk, “Modelling Player Preferences in AR Mobile Games” in *Proc. IEEE CoG*, 2019.

[2] J. Cohen, “Embedded Speech Recognition applications in Mobile Phones: Status, Trends, and Challenges” in *Proc. IEEE ICASSP* , 2008.

[3] Z. Zhou, *et al.*, “Social Big-Data-Based Content Dissemination in Internet of Vehicles” *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 768–777, 2018.

[4] S. Chen, J. Hu, Y. Shi, L. Zhao, and W. Li, “A Vision of C-V2X: Technologies, Field Testing, and Challenges With Chinese Development” *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3872–3881, 2020.

[5] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, “Connected Vehicles: Solutions and Challenges” *IEEE Internet Things J.*, vol. 1, no. 4, pp. 289–299, 2014.

[6] Q. Yuan, *et al.*, “Toward Efficient Content Delivery for Automated Driving Services: An Edge Computing Solution” *IEEE Netw.*, vol. 32, no. 1, pp. 80–86, 2018.

[7] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. ZHANG, “Mobile-Edge Computing for Vehicular Networks: A Promising Network Paradigm with Predictive Off-Loading” *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 36–44, 2017.

[8] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A Survey on Mobile Edge Computing: The Communication Perspective” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 2017.

[9] P. Mach and Z. Becvar, “Mobile Edge Computing: A Survey on Architecture and Computation Offloading” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 2017.

[10] J. Zhang, H. Guo, J. Liu, and Y. Zhang, “Task Offloading in Vehicular Edge Computing Networks: A Load-Balancing Solution” *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2092–2104, 2020.

[11] C. Li, *et al.*, “Parked Vehicular Computing for Energy-Efficient Internet of Vehicles: A Contract Theoretic Approach” *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6079–6088, 2019.

[12] C. Yang, Wei. Lou, Y. Liu, and S. Xie “Resource Allocation for Edge Computing-Based Vehicle Platoon on Freeway: A Contract-Optimization Approach” *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15988–16000, 2020.

[13] Y. Wang, Z. -Y. Ru, K. Wang, and P. -Q. Huang, “Joint Deployment and Task Scheduling Optimization for Large-Scale Mobile Users in Multi-UAV-Enabled Mobile Edge Computing” *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3984–3997, 2020.

[14] A. Islam and S. Y. Shin, "BUAV: A blockchain based secure UAV-assisted data acquisition scheme in Internet of Things" *J. Commun. Netw.*, vol. 21, no. 5, pp. 491–502, 2020.

[15] A. Islam and S. Y. Shin, "BUS: A Blockchain-Enabled Data Acquisition Scheme With the Assistance of UAV Swarm in Internet of Things" *IEEE Access*, vol. 7, pp. 103231–103249, 2019.

[16] A. Islam, T. Rahim, MD Masuduzzaman, and S. Y. Shin, "A Blockchain-Based Artificial Intelligence-Empowered Contagious Pandemic Situation Supervision Scheme Using Internet of Drone Things" *IEEE Wireless Commun.*, 2021.

[17] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone Small Cells in the Clouds: Design, Deployment and Performance Analysis" in *Proc. IEEE GLOBECOM*, 2015.

[18] C. Li, *et al.*, "Enhanced signalling provisioning for UAV-enabled MEC: A GWFRFT-based energy-spreading transmission approach" *IET Commun.*, vol. 14, no. 15, pp. 2524–2531, 2020.

[19] B. Hang, *et al.*, "A User Association Policy for UAV-aided Time-varying Vehicular Networks with MEC" in *Proc. IEEE WCNC*, 2020.

[20] Y. Zeng, R. Zhang, T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges" *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, 2016.

[21] L. Zhang, *et al.*, "Energy-Aware Dynamic Resource Allocation in UAV Assisted Mobile Edge Computing Over Social Internet of Vehicles" *IEEE Access*, vol. 6, pp. 56700–56715, 2018.

[22] S. Ahmed, M. Z. Chowdhury, and Y. M. Jang, "Energy-Efficient UAV Relaying Communications to Serve Ground Nodes" *IEEE Commun. Lett.*, vol. 24, no. 4, pp. 849–852, 2020.

[23] C. Zhan, H. Hu, X. Sui, Z. Liu, and D. Niyato, "Completion Time and Energy Optimization in the UAV-Enabled Mobile-Edge Computing System" *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7808–7822, 2020.

[24] Q. Liu, *et al.*, "Path Planning for UAV-Mounted Mobile Edge Computing With Deep Reinforcement Learning" *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5723–5728, 2020.

[25] N. A. Khofiyah, S. Maret, W. Sutopo, and B. D. A. Nugroho, "Goldsmith's Commercialization Model for Feasibility Study of Technology Lithium Battery Pack Drone" in *Proc. ICEVA*, 2018.

[26] M. C. Achtelik, J. Stumpf, D. Gurdan, and K. Doth, "Design of a flexible high performance quadcopter platform breaking the MAV endurance record with laser power beaming" in *Proc. IEEE/RSJ IROS*, 2011.

[27] E. Avila, *et al.*, "Energy Management of a Solar-Battery Powered Fixed-Wing UAV" in *Proc. INCISCOS*, 2018.

[28] R. A. Sowah, M. A. Acquah, A. R. Ofoli, G. A. Mills, and K. M. Koumadi, "Rotational Energy Harvesting To Prolong Flight Duration of Quadcopters" *IEEE Trans. Ind. Appl.*, vol. 53, no. 5, pp. 4965–4972, 2017.

[29] W. Zhang, Q. Ding, C. Zeng, and C. Wang, "Research on Duration Estimation of Rotor UAV Based on Flight Condition-Energy Consumption Identification" *J. Physics: Conference Series*, 2019.

[30] C. Wang, J. Wang, Y. Shen, and X. Zhang, "Autonomous Navigation of UAVs in Large-Scale Complex Environments: A Deep Reinforcement Learning Approach" *IEEE Trans. Veh. Technol.* vol. 68, no. 3, pp. 2124–2136, 2019.

[31] B. Shi, *et al.*, "Complex-Valued Convolutional Neural Networks Design and its Application on UAV DOA Estimation in Urban Environments" *J. Commun. Inf. Netw.*, vol. 5, no. 2, pp. 130–137, 2020.

[32] O. S. Oubbati, N. Chaib, A. Lakas, P. Lorenz, and A. Rachedi, "UAV-Assisted Supporting Services Connectivity in Urban VANETs" *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3944–3951, 2019.

[33] L. Zhao, *et al.*, "A Novel Cost Optimization Strategy for SDN-Enabled UAV-Assisted Vehicular Computation Offloading" *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3664–3674, 2021.

[34] L. Hu, *et al.*, "Ready Player One: UAV-Clustering-Based Multi-Task Offloading for Vehicular VR/AR Gaming" *IEEE Netw.*, vol. 33, no. 3, pp. 42–48, 2019.

[35] Q. Yang and S. Yoo, "Optimal UAV Path Planning: Sensing Data Acquisition Over IoT Sensor Networks Using Multi-Objective Bio-Inspired Algorithms" *IEEE Access*, vol. 6, pp. 13671–13684, 2018.

[36] T. Morita, K. Oyama, T. Mikoshi, and T. Nishizono, "Decision Making Support of UAV Path Planning for Efficient Sensing in Radiation Dose Mapping" in *Proc. IEEE COMPSAC*, 2018.

[37] K. Zhang, Y. Zhu, S. Leng and Y. He, S. Maharjan, and Y. Zhang, "Deep Learning Empowered Task Offloading for Mobile Edge Computing in Urban Informatics" *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7635–7647, 2019.

[38] Ji, Xiang, "Research on Path Planning Methods Using Multi-Feature Fusion for Optimizing Energy Consumption of Rotary-Wing UAVs", Northwest University, China, 2019.

[39] A. Al-Hourani S. Kandeepan, and S. Lardner, "Optimal LAP Altitude for Maximum Coverage" *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, 2014.

[40] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments" in *Proc. IEEE GLOBCOM*, 2014.

[41] G. Ke, *et al.*,"LightGBM: A Highly Efficient Gradient Boosting Decision Tree" in *Proc. NIPS*, 2017.

[42] "Maze game", https://gym.openai.com/envs/.

[43] "Traffic line access time prediction", https://js.dclab.run/v2/cmptDetail.html?id=175.

Zhiwei Wu received his B.E. degree from school of Electrical and Information Engineering, Anhui University Of Science & Technology (AUST), HuaiNan, China, in 2019. He is currently pursuing the M.S. degree with the school of Automation, Guangdong University of Technology, Guangzhou, China. His research interests include wireless communication networks, UAV cooperative communications, and intelligent edge computing.

Zilin Yang received the B.E. degree from the school of Automation, Guangdong University of Technology (GDUT), Guangzhou, China, in 2021. She is also a Member of the Guangdong Key Laboratory of IoT Information Technology, Guangdong University of Technology, Guangzhou. Her research interests include intelligent transportation, Internet of vehicles, and simultaneous localization and mapping technology.

Chao Yang received the Ph.D. degree in signal and information processing from the South China University of Technology, Guangzhou, China, in 2013. He is currently with the School of Automation, Guangdong University of Technology. From 2014 to 2016, he was a Research Associate with the Department of Computing, Hong Kong Polytechnic University. His research interest focuses on Internet of Vehicles, smart grid, and edge computing.

Jixu Lin received his B.E. degree from school of Electronic and Electrical Engineering, Zhaoqing University, Zhaoqing, China, in 2019. He is currently pursuing the M.S. degree with the school of Automation, Guangdong University of Technology (GDUT), Guangzhou, China. His research interests include bigdata analysis, UAV cooperative communications, and intelligent transportation.

Yi Liu received his Ph.D. degree from South China University of Technology (SCUT), Guangzhou, China, in 2011. After that, he joined the Singapore University of Technology and Design (SUTD) as a post-doctoral. In 2014, he worked in the Institute of Intelligent Information Processing at Guangdong University of Technology (GDUT), where he is now a Full Professor. His research interests include wireless communication networks, cooperative communications, smart grid and intelligent edge computing.

Xin Chen received the Ph.D. degree in Bioinformatics from Harbin Medical University, Harbin, China, in 2012. After that, she was a Postdoctoral Fellow at the Faculty of Health Sciences, University of Macau. From 2016, she works in the Institute of Intelligent Information Processing at Guangdong University of Technology (GDUT). Her research interest focuses on computational biology, wireless big data analysis and convex optimization.