Satellite-assisted UAV Trajectory Control in Hostile Jamming Environments

Chen Han, Aijun Liu, Kang An, Haichao Wang, Gan Zheng, *Fellow, IEEE*, Symeon Chatzinotas, *Senior Member, IEEE*, Liangyu Huo and Xinhai Tong

***Abstract*—Satellite and unmanned aerial vehicle (UAV) networks have been introduced as enhanced approach to provide dynamic control, massive connections and global coverage for future wireless communication systems. This paper considers a coordinated satellite-UAV communication system, where the UAV performs the environmental reconnaissance task with the assistance of satellite in a hostile jamming environment. To fulfill this task, the UAV needs to realize autonomous trajectory control and upload the collected data to the satellite. With the aid of the uploading data, the satellite builds the environment situation map integrating the beam quality, jamming status, and traffic distribution. Accordingly, we propose a closed-loop anti-jamming dynamic trajectory optimization approach, which is divided into three stages. Firstly, a coarse trajectory planning is made according to the limited prior information and preset points. Secondly, the flight control between two adjacent preset points is formulated as a Markov decision process, and reinforcement learning (RL) based automatic flying control algorithms are proposed to explore the unknown hostile environment and realize autonomous and precise trajectory control. Thirdly, based on the collected data during the UAV's flight, the satellite utilizes an environment situation estimating algorithm to build an environment situation map, which is used to reselect the preset points for the first stage and provide better initialization for the RL process in the second stage. Simulation results verify the validity and superiority of the proposed approach.**

***Index Terms*—Jamming, Satellite-UAV coordination communication, Trajectory optimization, Reinforcement learning, Graph theory.**

This paper appears in part at the 6th IEEE International Conference on Computer and Communications (ICCC 2020), Chengdu, China [1]. This work was supported in part by the National Natural Science Foundation of China under Grant No. 61671476, 62001514, in part by the Natural Science Foundation of Jiangsu Province under Grant No. BK20180578, in part by the National Key Research and Development Program of China under Grant No. 2018YFB1801103, and in part by the China Postdoctoral Science Foundation under Grant No. 2019M651648. *(Corresponding author: Aijun Liu.)*

Chen Han, Aijun Liu, Haichao Wang and Xinhai Tong are with the College of Communications Engineering, Army Engineering University, Nanjing 210007, China (e-mail: hanchen.lgd@gmail.com; liuaj.cn@163.com; whcwl0919@sina.com; tongxinhai2012@163.com).

Kang An is with the Sixty-Third Research Institute, National University of Defense Technology, Nanjing 21007, China (e-mail:ankang89@nudt.edu.cn).

Gan Zheng is with the Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Loughborough LE113TU, U.K. (e-mail: g.zheng@lboro.ac.uk).

Symeon Chatzinotas is with Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg, Luxembourg (symeon.chatzinotas@uni.lu).

Liangyu Huo is with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China (e-mail:huoliangyu@buaa.edu.cn).

I. Introduction

A. Motivations

Satellite communication systems (SCS) and unmanned aerial vehicles (UAV) have played significant roles in the field of wireless communications [2], [3]. When the terrestrial base stations (BS) are unavailable in the hostile areas, the assistance of SCS is essential for UAV to carry out the environmental reconnaissance and data collection task. Efficient trajectory control enables the UAV to achieve higher reconnaissance performance [4]. However, the trajectory control problem in the hostile environments faces the following challenges: i) With the assistance of SCS, beam selection will seriously affect the UAV trajectory planning. The beam quality and data collection should be jointly considered in the trajectory control process; ii) In the hostile environment, the threat of malicious jamming remains a severe challenge, which can significantly reduce UAV's uploading data; iii) Due to the uncertain environmental states and huge action search space, the overall information for trajectory control is difficult to obtain, and the previous and centralized approaches can not work in the hostile environment. Motivated by the above observations, this paper investigates the satellite-assisted UAV trajectory control problem in the hostile jamming environments.

B. Related Works

Due to the advantages of fast deployment and low cost, UAV is believed to be a potentially promising choice for data collection and emergency communications [5]. The surging UAV applications inspire researchers to model and analyze UAV communication patterns [6]. In the hostile environment, the terrestrial BS is unable to provide access service to the UAV, thus the UAV has to rely on information support from the SCS, especially low earth orbit (LEO) SCS [7]. However, due to the long distance and large delay, it is difficult for the LEO SCS to perform near-surface and low latency reconnaissance tasks. Due to their respective limitations, SCS and UAVs can be combined to form a coordinated satellite-UAV network [8]. The coordination between satellites, UAVs, and terrestrial systems has drawn increasing attention. The authors in [9] studied a satellite-UAV mobile edge caching IoT system, in which UAV was used to enhance the data cashing ability. The authors in [10] studied the statistical framework of QoS requirements for airborne wireless networks, including satellites, airships, and UAVs. In [11], multiple satellites cooperatively served the UAVs and mobile terminals, and NOMA technology was used to access satellite networks. In [12], the authors investigated

the integration of SCS and aerial communications for UAVs and other flying vehicles. In [13], the authors solved the overload problems of ground BS by using UAV as a mobile BS and optimizing the bandwidth, users' partitioning, and UAV's trajectory. However, the existing works mainly focus on the cooperative scenarios between the SCS and terrestrial systems or UAV and terrestrial systems, while there exists few works on cooperative communications between satellites and UAVs.

UAV trajectory control is an important issue to improve the system performance and data quality, which is closely related to the UAV's kinematic parameters [14]. In [15], the authors designed a coarse trajectory for energy minimization considering UAV's velocities and latency constraints. The UAV trajectory optimization was studied in [16] which can learn the environment characteristic quickly. In [17], the authors studied UAV trajectory optimization at the edges of three neighboring cells to offload data for BSs. The authors in [18] investigated an efficient UAV-enabled power transfer scheme via trajectory design. The authors in [19] proposed a decoupled heuristic solution to minimize the total time spent on travelling and hovering for UAV flight. In most works, the data collection points are determined and known to UAVs in advance [17], [20], [21]. However, the uncertain data collection points are rarely investigated in the existing works, which is a challenge that must be faced in the hostile jamming environments.

Joint optimization problems for UAV communications have also been studied extensively. The authors in [22] optimized the UAV trajectory and in-flight power to enhance the coverage of hybrid satellite terrestrial networks. The authors in [23] investigated the joint optimization problem of power control and trajectory design in UAV-aided hybrid satellite terrestrial networks. In [8], the authors studied multi-domain resource allocation for the coordinated satellite-UAV network to improve the performance of wide-area massive access. The authors in [20] proposed a new design framework to jointly optimize the UAV energy consumption and communication throughput by UAV trajectory control. Collaborative multi-agent task is also an important research direction in the trajectory design of UAV [24], [25]. The authors in [21] investigated the multi-UAV communication scheduling and the joint optimization problem of power control and trajectory design. On the one hand, the joint optimization works considering the anti-jamming requirement are rarely investigated in the state-of-the-art. In addition, beam switching is also a condition that seriously affects UAV trajectory planning, because UAVs cannot complete data uploading in the area with a poor beam quality. Thus, the beam selection and trajectory control should be jointly optimized rather than in a separated or partial optimization pattern [18], [21], especially in the satellite-UAV coordination networks.

The above-mentioned works mainly solved the trajectory design by centralized optimization rather than decentralized planning and automatic adjustment, and these works mainly makes pre-deployment before flight. However, it should be noted that the global information is generally difficult to obtain in the hostile environments, which needs to take all of the environment situation into consideration. Thus, it is unavailable to apply the centralized planning approach. However, reinforcement learning (RL) approach is able to grasp the uncertain environment by the exploration-exploitation mechanism, which can be used to adjust flying trajectory automatically. RL algorithm has been widely used in the field of wireless communications. In [26], Q-Learning was used to analyze spectrum availability and make optimal anti-jamming policies. An actor-critic deep RL approach was proposed in [27], [28] to solve the time-slot allocation problem in UAV-Aided Networks. The authors in [29] used RL algorithm for anti-jamming defense across multiple regions. Besides, Q-learning was used in [30] to form dynamic anti-jamming coalition for the devices of satellite-enabled army internet of things. In [31], the authors utilized RL scheme to solve the anti-jamming routing problem for internet of satellite. In [32], Q-Learning was used to deal with the joint anti-jamming problem of routing selection, channel allocation and power control. In this paper, RL is applied to realize the automatic UAV flight, obtain the situation information of hostile environment, and perform effective anti-jamming defense. Differently from our previous work [1], which only considered point-to-point automatic flight in a hostile environment, this paper investigates a closed-loop complete trajectory optimazation problem, including coarse trajectory design, point-to-point fine trajectory control and environment situation mapping.

C. Contributions and Organizations

In this paper, considering the challenges of uneven traffic distribution, unknown jamming state and uncertain beam quality, the satellite-assisted UAV trajectory control problem is decoupled into three closed-loop sub-problems for effective solutions. The main contributions are summarized as follows:

- The satellite-assisted UAV trajectory control problem is first formulated. Specifically, the UAV performs data collection and environment mapping task with the assistance of satellite, and it attempts to improve data quality and transmission performance by trajectory control.
- The initial trajectory control problem is decoupled into three closed-loop sub-problems, namely, preliminary planning of coarse trajectory, point-to-point precise trajectory control and environmental situation assessment. Consequently, we propose an anti-jamming dynamic trajectory optimization approach to deal with the challenges of uncertain traffic data, unknown jamming state and uneven beam quality and to achieve efficient trajectory optimization;
- Computational complexity, convergence analysis and optimality analysis are given in details. Simulation results validate the effectiveness of the proposed approach, and show the influence of different factors regarding the algorithm performance.

The remaining part of this paper is organized as follows. Section II gives the system model and problem formulation. The proposed anti-jamming dynamic trajectory optimization approach is elucidated in Section III. Simulation results and discussion are shown in Section IV. Section V draws the conclusions.

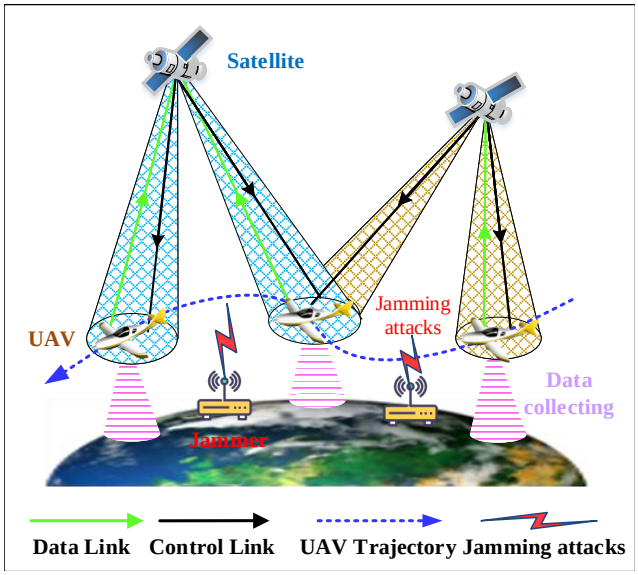


Fig. 1: The anti-jamming trajectory control scene. The UAV performs environmental reconnaissance and data collection task with the assistant of LEO SCS in the hostile jamming environment.

II. System Model and Problem Formulation

A. System Model

1) Task model: The anti-jamming trajectory control scene for UAV is shown in Fig. 1. The large UAV, equipped with electronic reconnaissance equipment, is specially used for intercepting and recording enemy electromagnetic radiation signals from the ground [33]. Then, the reconnaissance data is uploaded to satellites for processing and analyzing. The UAV is able to achieve automatically trajectory control with the assistance of satellites[1]. The satellite provides access channels to UAV, and transmits the environmental situation assessment to the UAV via downlink. Meanwhile, the UAV collects data in the hostile jamming environment and uploads collected data to the satellite via the uplink. However, ground-based jammers in hostile environments pose a serious threat to UAV flight. It will launch jamming attacks to disrupt the UAV's data uploading and hinder the UAV's in-flight reconnaissance task. Thus, UAV attempts to improve data quality and anti-jamming performance by trajectory control.

In this paper, the target are is meshed into grids [33]. The size of each grid is set according to the coverage range of a spot beam. Each grid's position is fixed and indexed by i. We consider that the hot grids show aggregation characteristics, where UAV may collect larger traffic data [34]. In the i-th grid, the collected traffic data is $D(i,t)$ at time t, which follows the Poisson distribution with a Poisson parameter $\lambda_{i,t}$ [35]:

$$\Pr\{D(i,t)=x\}=\frac{(\lambda_{i,t})^{x}}{(x)!}e^{-\lambda_{i,t}}, \tag{1}$$

where $\Pr\{D(i,t)=x\}$ is the probability that this grid generates x bits data. The traffic of each grid is generated randomly according to the local traffic law. $\lambda_{i,t}^{h}$ is regarded as the mean traffic of high traffic grids, and the other's Poisson parameter is set as $\lambda_{i,t}^{l}$. Moreover, $\lambda_{i,t}$ is diverse in different grids and varies over time, and it is only related to the local traffic law of each grid. As shown in Fig. 2(a), the traffic distribution in different grids is uneven, and the hot points have larger traffic data [30]. UAV need to reach as many hot points as possible during the autonomous flying. However, as shown in Fig. 2(b)-(d), UAV cannot upload data in a grid with poor beam quality, even in a hot point. Thus, data collection and beam quality should be considered comprehensively in trajectory control, and the UAV tends to access the grids with large traffic and higher signal-to-jamming-plus-noise ratio (SJNR)[2], which means good signal strength and a low probability of jamming.

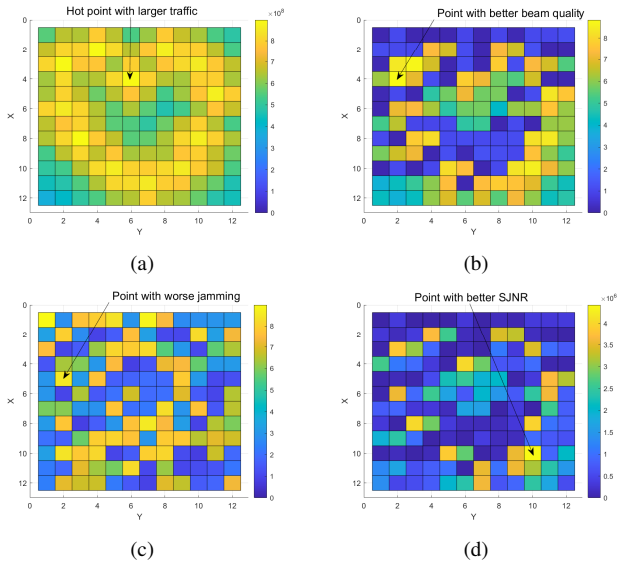


Fig. 2: The environment setting obtained by averaging 1000 trials. (a) Traffic distribution setting; (b) Beam quality setting; (c) Jamming status setting; (d) The corresponding SJNR.

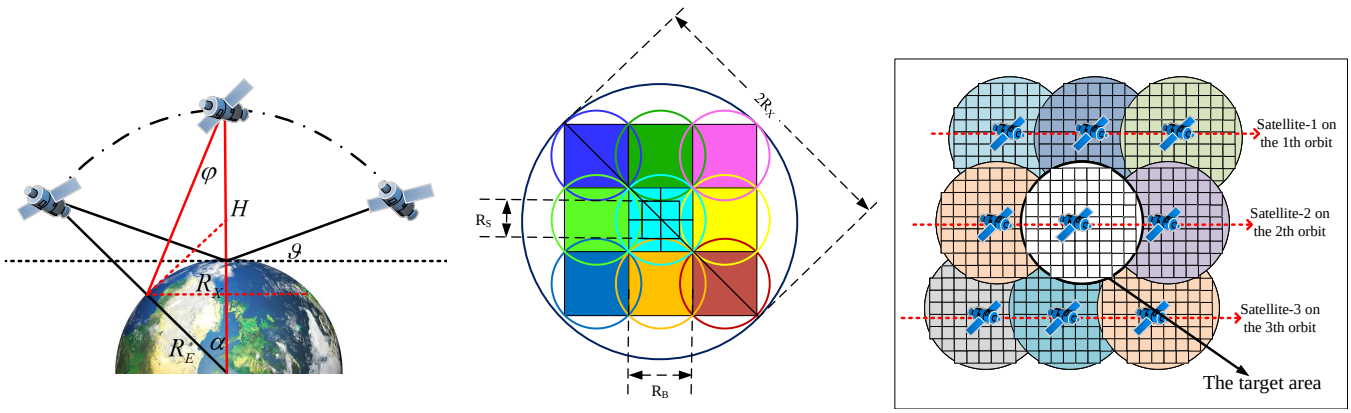
2) Satellite beam model: The LEO satellite constellation completes the global coverage with an orbit height of H_s, which consists of N_o circle polar orbits with each orbit placing N_s satellites. As shown in Fig. 3(a), the earth radius is R_E, and the minimum elevation angle from ground observation point to the satellite is ϑ. By using geometrical relationship, the lower half angle of maximum visibility is $\varphi=\arcsin\left[\cos\vartheta\times R_E/\left(R_E+H_s\right)\right]$. The geocentric angle corresponding to φ is:

$$\alpha=\arccos\left[\cos\vartheta\times R_E/\left(R_E+H_s\right)\right]-\vartheta. \tag{2}$$

The coverage radius of single satellite is $R_X=R_E\sin\alpha$, and the linear velocity and angular velocity of satellite are $v_l=\sqrt{GM/\left(R_E+H_s\right)}$ and $v_a=v_l/\left(R_E+H_s\right)$ respectively, where G and M are the gravitational constant and earth's

[1] Note that this task is not a one-time mission, but one that will obtain situational information of the target area through multiple reconnaissance flights over a period.

[2] Although we can set different distribution maps of traffic, beam quality, and jamming state, in general, the jammers will be deployed close to the important grids with high flow, and in these grids, the quality of our satellites is usually poor. Of course, we can normalize the importance of grids by setting different weights, which is also applicable to the proposed approach below.



(a) (b) (c)

Fig. 3: (a) The satellite movement model; (b) The satellite beam model which is simplified to square beam; (c) The coverage model of LEO satellite constellation.

mass. The longest continuous coverage time of a satellite to a user is:

$$T_c = 2\alpha / v_a. \tag{3}$$

Satellites use directional antennas and their transmitted signal is modeled using a beam. The square beam shown in Fig. 3(b) is used to illustrate the circular beam for simplified calculation. The satellite coverage are is meshed into $N_B \times N_B$ beam squares, and the length of each beam square is approximated as:

$$R_B = \sqrt{2} R_X / N_B. \tag{4}$$

The coverage model of LEO satellite constellation is shown in Fig. 3(c). As satellites sweeping the covered area, the user needs to switch frequently between the different beam cells or even different satellites. The switching time per beam in the direction of satellite moving is:

$$T_b = T_c / N_B. \tag{5}$$

The minimum distance required for beaming switch is:

$$R_S = (2 - \sqrt{2}) R_B. \tag{6}$$

3) Transmission channel model: According to [36], the shadowed-Rician model is a popular satellite link model with a significantly reduced computational complexity to describe the accurate amplitude fluctuation of signal envelope. Consequently, the probability density function of the channel gain of the transmission link from the i-th grid to the satellite (i.e., h^i) is expressed as $Y_1 = \left|h^i\right|^2$:

$$f_{Y_1}(x) = \alpha_i \exp(-\beta_i x) {}_1F_1(m_i; 1; \delta_i x), \tag{7}$$

where ${}_1F_1(m_i; 1; \delta_i x)$ is the confluent hypergeometric function, $\beta_i = 1/2\chi_i$, $\alpha_i = (2\chi_i m_i / (2\chi_i m_i + \mu_i))^{m_i} / 2\chi_i$, $\delta_i = \mu_i / 2\chi_i (2\chi_i m_i + \mu_i)$. $m_i \in (0, \infty)$ is the Nakagami-m parameter, μ_i is the mean power of the line-of-sight (LOS) component, $2\chi_i$ is the mean value of the multi-path power.

Ground-based jammers jammers are randomly distributed in the target region. If a jammer lies in the i-th grid, i.e., $\delta(J_i = 1) = 1$, it launches jamming attacks towards UAVs according to jamming probability of Pr_d. According to [37], Nakagami-m fading distribution is used to model the air-ground links, including the data collecting links and the jamming links. The probability density function of the channel gain of the jamming link in the i-th grid (i.e., h_J) is defined as $Y_2 = |h_J|^2$:

$$f_{Y_2}(x) = \frac{x^{\theta_i - 1} \exp(-\frac{x}{\eta_i})}{\Gamma(\theta_i) {\eta_i}^{\theta_i}}, \tag{8}$$

where $\Gamma(\cdot)$ is the Gamma function, and the average power is $\eta_i \theta_i$.

UAV is equipped with a single antenna, while satellites are equipped with multiple antennas. The received signal at the satellite is:

$$\begin{aligned} y_i(t) &= \sqrt{G_s G_u} {d_i}^{-\beta} h_k^{i,t} s_i(t) e^{j2\pi f_d^k(t)t} + \sigma^2 + \sigma_B^2 + \sigma_J^2, \\ \sigma_B^2 &= \sum_{b \neq k} \sqrt{G_s G_u} {d_i}^{-\beta} h_b^{i,t} s_i'(t) e^{j2\pi f_d^b(t)t}, \\ \sigma_J^2 &= Pr_d \delta(J_{i,t} = 1) d_J^{-\beta} h_J p_J, \end{aligned} \tag{9}$$

where $s_i(t)$ is the signal transmitted from UAV in the i-th grid at time t. $E[|s_i(t)|^2] = p_U$, and p_U is the UAV transmitting power. G_s is the antenna gain at satellite, G_u is the antenna gain at UAV. d_i is the transmission distance from the i-th grid to satellites [38]. β is the large-scale fading coefficient. $h_k^{i,t}$ is the channel gain using the k-th beam. σ^2 is the channel noise, and σ_B^2 is the inter-beam interference introduced by other beams, $b \neq k$. The Doppler shift caused by the satellite movement which is assumed to be fully compensated by carrier frequency offset and correction [39]. G_s, d_i, G_u are all constants, thus, without loss of generality, we set $\sqrt{G_s G_u} {d_i}^{-\beta} = 1$ [38]. The maximum transmission rate of the satellite uplink is defined as:

$$C(i,t) = B_{i,t} \log_2(1 + r_{i,t}) \delta(r_{i,t} \geq r_{th}), \tag{10}$$

$$r_{i,t} = \frac{h_k^{i,t} p_U}{Pr_d \delta(J_{i,t} = 1) h_J d_J^{-\beta} p_J + \sigma^2 + \sigma_B^2}, \tag{11}$$

where $r_{i,t}$ denotes the SJNR at the satellite receiver from the i-th beam cell at time t, and p_J represent the jamming

power. $B_{i,t}$ represents the maximum access channel bandwidth provided by the i-th beam cell at time t. When there is no channel remaining in this beam cell, i.e., $B_{i,t} = 0$, or the channel fading of this beam is very severe, i.e., $r_{i,t} < r_{th}$, the UAV cannot upload data, i.e., $\delta(r_{i,t} \geq r_{th}) = 0$. Otherwise, this beam cell could provide available service, and $\delta(r_{i,t} \geq r_{th}) = 1$.

4) UAV movement model: We consider a 3D coordinate system where the UAV flies at a fixed altitude H_u with a speed of v_u. To reduce the computational complexity and implement the coarse trajectory design, the Manhattan network model is used to model the meshed target region. The target area is set as $N_B \times N_B$ meshed grids. The grid is set by jointly considering the satellite beam coverage and switching strategy. Each grid represents a beam cell, which will be covered by another beam at every T_b. The coordinates projected on the horizontal plane of the location at current moment and next moment are defined as $l_t = [x_t, y_t]$ and $l_{t+1} = [x_{t+1}, y_{t+1}]$. UAV can choose four motion selection: forward, backward, left, and right [33]. Then, UAV goes into the adjacent grid, limited by $\|l_t - l_{t+1}\| = 1$, where $\|l_t - l_{t+1}\|$ represents the Manhattan distance from l_t to l_{t+1},

$$L_{l_t,l_{t+1}} = \|l_t - l_{t+1}\| = |x_t - x_{t+1}| + |y_t - y_{t+1}|, \quad x \in [1, N_B], y \in [1, N_B]. \tag{12}$$

B. Problem Formulation

Energy consumption is an important factor for UAV trajectory optimization. The total energy consumption of the UAV includes two components [19]. The first one is the propulsion energy, which is required for supporting its mobility. The other component is the data uploading energy, which is mainly related to the uploading data and corresponding channel state.

As remarked in [15], [40], the energy consumption is related to UAV's velocity and flying time. However, in this paper, the UAV's velocity is fixed, thus, the energy consumption is positively correlated with the flight distance. To simplify the calculation, the propulsion energy consumption that the UAV flies from grid i to grid j is defined as:

$$S^v_{ij} = e_1 L_{ij}, \tag{13}$$

where e_1 is a constant, and L_{ij} is the Manhattan distance that the UAV passes through from grid i to grid j.

The current location of the UAV is l_t, and the data uploading energy consumption in this grid is:

$$S^c_{l_t} = e_2 \hat{D}_{l_t}, \tag{14}$$

where $\hat{D}_{l_t}$ is the uploading data traffic at l_t-th grid. In our assumption, UAV's transmission power is constant, and the energy consumption of data uploading is actually corresponding to the during of data uploading. Thus, it is related to the amount of uploading data. Without loss of generality, we set it to a constant regarding to per uploading bit which is represented by e_2. In addition, location and link state affect the amount of data that can be uploaded. UAV has a cache stack with a maximal capacity of $|\mathcal{K}|$ for temporarily storing collected data. If the current collected data is not fully uploaded to the satellite, the rest will be stored in the cache stack waiting for the next transmission. Moreover, data will be discarded if it has been cached for a long time. At time $t+1$, the location of UAV is l_{t+1}. At the beginning when the UAV flies into l_{t+1}, the cached data $\mathcal{K}(l_{t+1})$ is:

$$\mathcal{K}(l_{t+1}) = \max\left\{0, \min\left\{\mathcal{K}(l_t) + D(l_t, t) - \hat{D}_{l_t}, |\mathcal{K}|\right\}\right\}, \tag{15}$$

where $D(l_t, t)$ represents the collected data in the l_t-th grid at time t. Then $\hat{D}_{l_t}$ is:

$$\hat{D}_{l_t} = \min\left\{\sum_{g=1}^{\lfloor T_{l_t}/T_b \rfloor} T_b \bar{C}(l_t, g), \mathcal{K}(l_{t-1})\right\}, \tag{16}$$

$$T_{l_t} \in \left\{\frac{R_S}{v_u}, \frac{R_B}{v_u}, \frac{\sqrt{2}R_B}{v_u}\right\}, \tag{17}$$

where T_{l_t} is the duration time in the l_t-th grid which can take three values, correspond to the three cases of poor beam quality, average beam quality, and good beam quality respectively. $\lfloor T_{l_t}/T_b \rfloor$ is the beam switching counts, $\sum_{g=1}^{\lfloor T_{l_t}/T_b \rfloor} T_b \bar{C}(l_t, g)$ is the corresponding total uploading data after multiple beam switching, where $\bar{C}(l_t, g)$ is the average transmission rate in the l_t grid in a period of time g.

The UAV tries to upload the maximum data under the constraint of energy consumption,

$$P: \max_{[l_1, l_2, \ldots l_t, \ldots, l_1]} \sum_t \hat{D}_{l_t}, \tag{18}$$
$$s.t. \quad \sum_t \left(S^v_{l_{t-1}, l_t} + S^c_{l_t}\right) \leq \mathbb{E}_0,$$

where $[l_1, l_2, \ldots l_t, \ldots, l_1]$ is the closed-loop flight trajectory, and $\mathbb{E}_0$ is the total energy.

Due to the uncertainty of the hostile environment, problem P is difficult to address via a centralized solution, not to mention that the accurate $\hat{D}_{l_t}$ is difficult to obtain, due to the uneven traffic distribution, unknown jamming stage and uncertain beam quality. Meanwhile, the increasing network size will greatly increase the computational complexity.

Remark 1: Dividing the problem P into two-step problems, $P1, P2$ can effectively reduce the computational complexity.

Assuming that the global information is known, and the modified Dijkstra algorithm is adopted, its computational complexity is $\mathcal{O}(N^2)$, where N is the grids number. Then, if we use the two-step method: firstly, Z preset points are selected and the coarse trajectory design is performed; secondly, perform point-to-point flying autonomously. Then, the computational complexity has been reduced to $\mathcal{O}\left(N^2/(Z-1)\right)$. Although the global information is not known in this paper, it can be expected that the overall computational complexity will be significantly reduced by the two-step approach. Therefore, the problem P can be divided into $P1$ and $P2$ to reduce computational complexity. As described below, $P1$ is used to select the preset points and design the coarse trajectory, thus dividing the target region into smaller regions according to adjacent preset points. Then, $P2$ is used to solve the autonomous trajectory planning problem between preset points.

1) Preliminary planning of coarse trajectory ($P1$): The preliminary planning of coarse trajectory can be formulated as a traveling salesman problem (TSP) [15], [41], [42]. It targets finding the shortest path visiting all the points in different locations exactly once. In this sub-problem, the UAV needs to find a designed trajectory visiting all the preset points $l_p = [l_p^1, l_p^2, \ldots, l_p^N]$ and flies back to the starting point with the lowest flying energy cost. Specifically, the UAV first needs to initialize a preliminary rough trajectory $\xi = \Phi\left(l_p^1, l_p^2, \ldots, l_p^N\right)$ and gradually modifies this trajectory to minimize the expected cost. Φ represents a rearrangement transformation.

Let w_{ij} be a binary decision variable which is equal to 1 if the UAV flies from preset point l_p^i to l_p^j, and 0 otherwise. The objective is to minimize the total propulsion energy consumption while visiting all preset points,

$$P1: \min_{\xi=\Phi\left(l_p^1, l_p^2, \ldots, l_p^N\right)} \mathbb{E}^v, \tag{19}$$

$$s.t. \quad \mathbb{E}^v = \sum_{l_p^i \in l_p} \sum_{l_p^j \in l_p, j \neq i} w_{ij} S_{ij}^v, \tag{19a}$$

$$\sum_{l_p^j \in l_p} x_{ij} = 1, \forall l_p^i \in l_p, \tag{19b}$$

$$\sum_{l_p^j \in l_p} x_{ji} = 1, \forall l_p^i \in l_p, \tag{19c}$$

where (19b) and (19c) ensure that each point can only be accessed once.

2) Point-to-point precise trajectory control ($P2$): In the meshed area, the selection of the next grid is not only required by data collection, but also limited by beam quality. However, in the hostile environment, only the stage information from the adjacent grid is observable. Furthermore, the next grid selection is only relevant to the current grid and the available action space, not to the previous selection. Thus, to realize the automatic exploration of the uncertain hostile environment, the point-to-point precise trajectory control problem from the n-th preset point ξ_n to the next ξ_{n+1} is formulated as a Markov decision problem (MDP) with state space S, action space A, reward function R [31]. The state space is defined as the current location:

$$\left\{s_k \in S \,|\, S = \xi_n, l_1^n, \ldots, l_k^n, \ldots, l_{\mathbb{K}}^n, \xi_{n+1}\right\}, \tag{20}$$

where ξ_n and ξ_{n+1} belong to the preset points set, $\xi_n, \xi_{n+1} \in \xi$, $\xi = \Phi\left(l_p\right)$, and they are the source point and destination point of this point-to-point trajectory control problem. l_k^n is the intermediate point between the source and destination points, and $\mathbb{K} = X_n \times Y_n$, $X_n \leq N_B$, $Y_n \leq N_B$ is the total grid number of the sub-network with ξ_n and ξ_{n+1} as vertices.

The action space in this paper is defined as:

$$\left\{a_k \in A \,|\, A = a_1, a_2, a_3, a_4\right\}, \tag{21}$$

where a_1, a_2, a_3, a_4 are expressed as the action of "Forward", "Backward", "Left", and "Right".

The reward function is defined as the traffic data uploaded in the current grid l_k^n:

$$R(l_k^n) = \hat{D}_{l_k^n}. \tag{22}$$

Due to the declining beam quality or the jamming impact, UAV needs to frequently switch beam grid and get into the grid with better channel state. The second sub-problem is that the UAV needs to maximize uploading data $\sum_{n=1}^{N-1} \mathbb{R}_n$ through the trajectory control in a piecewise way,

$$P2: \max_{\{\psi_n\}_{n=1}^{N-1}} \sum_{n=1}^{N-1} \mathbb{R}_n, \tag{23}$$

$$s.t. \quad \mathbb{R}_n = \left(R\left(\xi_n\right) + \sum_{k=1}^{K_n} R\left(l_k^n\right) + R\left(\xi_{n+1}\right)\right), \tag{23a}$$

$$\psi_n = \left[\xi_n, l_1^n, l_2^n, \ldots l_k^n \ldots l_{K_n}^n, \xi_{n+1}\right], \tag{23b}$$

$$\xi_n, \xi_{n+1} \in \xi, n \leq N-1, \tag{23c}$$

$$K_n \leq L_{\xi_n, \xi_{n+1}}^{th}, \tag{23d}$$

where $\psi_n = \left[\xi_n, l_1^n, l_2^n, \ldots l_k^n \ldots l_{K_n}^n, \xi_{n+1}\right]$ is the UAV precise trajectory from source ξ_n to the destination ξ_{n+1}. Due to the energy reserve, UAV can fly up to $L_{\xi_n, \xi_{n+1}}^{th}$ between the preset points ξ_n and ξ_{n+1}, i.e., $L_{\xi_n, \xi_{n+1}} \leq L_{\xi_n, \xi_{n+1}}^{th}$. Finally, according to $P1$ and $P2$, we can obtain a closed-loop flight trajectory, i.e., $[l_1, ..., l_t, ..., l_1] = [\xi_1, l_1^1, l_2^1 \ldots \xi_2, l_1^2, l_2^2 \ldots \xi_3 \ldots \xi_N \ldots \xi_1]$.

3) Environmental assessment and preset points selection ($P3$): The current environmental situation is assessed according to the historical data and satellite sensing results,

$$\widetilde{\mathbb{D}}(l_p^i) = \mathrm{E}\left[D\left(l_p^i, t\right)\right]_{\Pi}^t, \tag{24}$$

$$\widetilde{\mathbb{C}}(l_p^i) = \mathrm{E}\left[r\left(l_p^i, t\right)\right]_{\Pi}^t, \tag{25}$$

where E is an operation of segment averaging,

$$\mathrm{E}\left[Y(t)\right]_{\Pi}^t = \frac{1}{\Pi} \sum_{x=t-\Pi+1}^{t} Y(x). \tag{26}$$

where Π is the truncated length to improve the adaptability to environmental changes.

Then, in order to select the grids that can upload more data, $N < N_B^2$ preset points $l_p = [l_p^1, l_p^2, \ldots, l_p^N]$ are selected as the new preset points for the next flight,

$$P3: \max_{\{l_p^i\}_{i=1}^N} \sum_{i=1}^{N} \min\left\{\widetilde{\mathbb{D}}\left(l_p^i\right), \widetilde{\mathbb{C}}\left(l_p^i\right) T_{l_p^i}\right\}. \tag{27}$$

Meanwhile, the assessment results are used to provide better initialization of Q value for the RL approach.

Remark 2: With formulation of $P3$, the three-step problems, i.e., $P1, P2, P3$, will form a closed loop by means of iterative optimization and obtain the sub-optimal solution.

There is a high correlation between the grids in our environment setting. In a certain local area, there are only finite hot grids which can generate high traffic data. Because the hot grids show aggregation characteristics, if on a given flight, the UAV finds a better grid with more uploading data, it will gradually approach the extreme point of the local area, and it finally converges to the local optimum. In the subsequent flights, environmental estimation is further optimized based on the collected data. Meanwhile, the preset points selection is constantly optimized according to the environment situation assessment. Thus, the performance of the proposed approach

could be continuously improved according to the Pareto criterion. Therefore, the proposed closed-loop scheme can converge to a sub-optimal solution at least. As the number of flights increases, the UAV will have more accumulated knowledge of the environment, and the optimized trajectory will gradually move towards the sub-optimal solution.

In summary, the original problem P is decomposed into three closed-loop sub-problems: preliminary planning of coarse trajectory ($P1$), the sub-problem of point-to-point precise trajectory control ($P2$), and the sub-problem of environmental assessment and preset points selection ($P3$).

Specifically, to address the problem P, the UAV firstly designs a preliminary planning of coarse trajectory ($P1$); Next, it makes a point-to-point autonomous precise flight control ($P2$); Then, based on the collected data, $P3$ is formulated to assess the environment situation, and further optimize the selection of preset points for $P1$, and provide better initialization for the $P2$.

Additionally, in the initial stage, the UAV has no prior details about the hostile environment, thus, only satellite observation can be used to determine some preset points to plan the coarse trajectory. The UAV flies according to the preset trajectory but chooses a specific path autonomously between two preset points based on the actual situation and knowledge obtained by the environmental assessment.

Remark 3: The actual solution to the trajectory planning problem in a hostile environment studied can be composed of the following four stages:

(1) Build the hostile environment modeling based on existing data and experience;

(2) Off-line training under the reinforcement learning paradigm;

(3) Actual flight control according to the trained trajectory control strategy, meanwhile, collect the data during flight;

(4) The new collected data is used to further improve environment modeling and optimize the learning model to get better strategies.

Each actual flight provides further experience and a better strategy for the next flight. In this paper, reinforcement learning is used for offline learning training, which is indicated in problem $P2$ and Algorithm 3 and 4. Through problem $P3$ and Algorithm 5, the existing collected data is used for improving hostile environment modeling, and spectrum situation map is drawn for better learning and training.

III. Anti-jamming Dynamic Trajectory Optimization Approach

As shown in Fig. 4, the proposed anti-jamming dynamic trajectory optimization algorithm, which is shown in Algorithm 1, consists of three sub-algorithms: the coarse trajectory planning algorithm (Algorithm 2), the automatic flying control algorithm (Algorithm 3), and the environment situation estimating algorithm (Algorithm 5). We will describe these algorithms in detail as follows.

A. Coarse trajectory planning algorithm (Algorithm2)

Due to the lack of prior knowledge of the environment, the UAV first needs to initialize a preliminary coarse trajectory

Algorithm 1: Anti-jamming dynamic trajectory optimization algorithm (ADTO)

Input: The target region, and the initial satellite sensing results.

Output: The optimized trajectory $\{\psi_n^*\}_{n=1}^{N-1}$, and the environment situation assessment Θ.

Initialize the preset data points l_p^0 and the default trajectory $\{\psi_n^0\}_{n=1}^{N-1}$. Initialize the environment situation assessment $\Theta = 0$.

for *Each UAV flight* **do**

  Execute Algorithm 2, and obtain the preliminary planning of rough trajectory $\xi_f = \Phi(l_p)$.

  Execute Algorithm 3, and obtain the precise flying trajectory $\{\psi_n^f\}_{n=1}^{N-1}$, and calculate the corresponding total uploading data $\sum_{n=1}^{N-1} \mathbb{R}_n^f$.

  If the new trajectory obtains more benefits, update the trajectory, otherwise, maintain the original trajectory.

  Execute Algorithm 5, assess the environment situation Θ based on the collected data during flight.

  Reselect the preset data points l_p for the Algorithm 2 and provide better initial Q for Algorithm 3.

end

$\xi = \Phi\left(l_p^1, l_p^2, \ldots, l_p^N\right)$ and gradually modifies this trajectory in order to minimize the expected energy cost. Meanwhile, it is also a requirement to deal with the challenge of increasing scale of grids networks. The preliminary coarse trajectory problem, i.e., $P1$, is modeled as a TSP problem which is a well-known NP hard problem. TSP problem aims to find a Hamilton graph with a minimum weight for all the preset points $l_p = [l_p^1, l_p^2, \ldots, l_p^N]$, which are determined by raw satellite observations. Different from traditional dynamic programming whose computational complexity will become unacceptable as the increasing points, this paper utilizes a simulated annealing (SA) based search approach. At the current temperature T_e, a new trajectory $\hat{\xi}$ is generated by random perturbation transform Υ,

$$\hat{\xi} = \Upsilon\left(\xi\right), \tag{28}$$

where Υ consists of three patterns: randomly switching the order of two points, shifting the order of points, and inverting the intermediate points.

Then, the increment of the new trajectory $\Delta(\hat{\xi} \leftarrow \xi)$ is calculated as,

$$\Delta(\hat{\xi} \leftarrow \xi) = \mathbb{E}^v(\hat{\xi}) - \mathbb{E}^v(\xi). \tag{29}$$

If the new trajectory has lower energy consumption, then, the new trajectory is updated as the current trajectory. Otherwise, the acceptance probability Pr_e of the new trajectory is calculated:

$$Pr_e = \exp(-\frac{\Delta(\hat{\xi} \leftarrow \xi)}{T_e}). \tag{30}$$

According to the greedy criterion, a random probability value

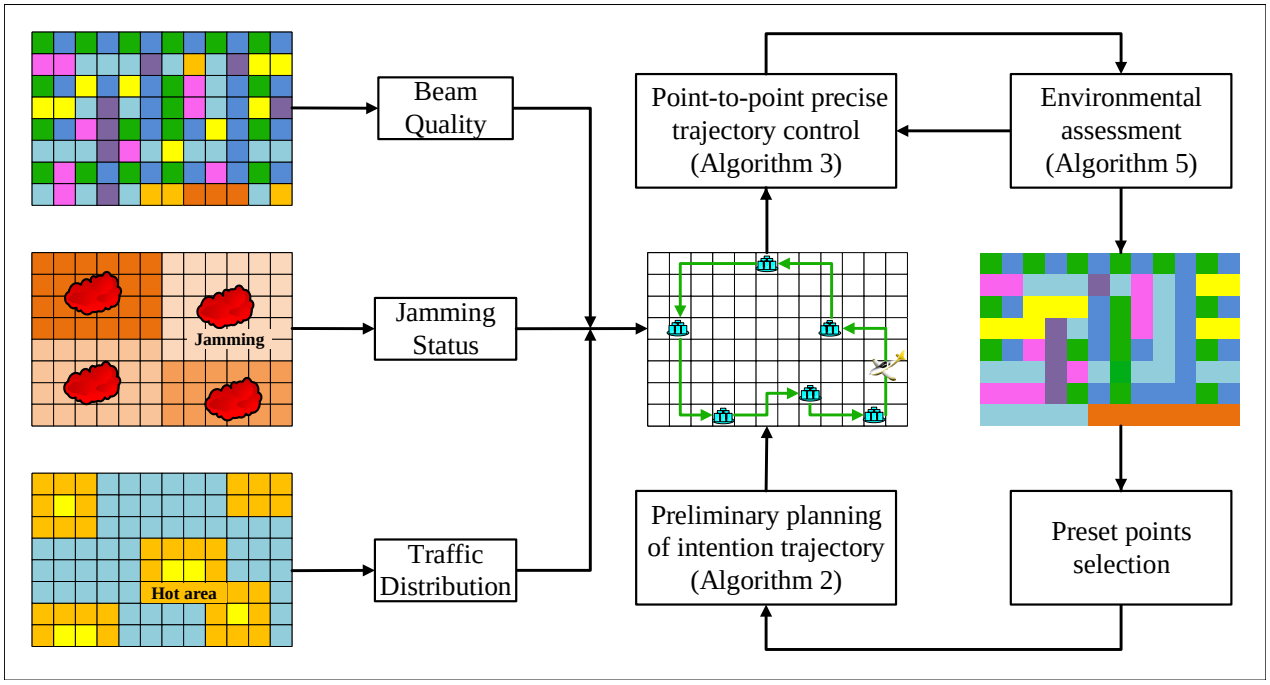


Fig. 4: The UAV trajectory control framework. In a hostile environment, the beam quality, jamming status, and traffic distribution of the current region are uncertain to UAV. To address this problem, the UAV firstly designs a preliminary planning of coarse trajectory (Algorithm 2); Next, it makes a point-to-point autonomous precise flight control (Algorithm 3); Then, based on the collected data, satellite could assess the environment situation, and further optimize the selection of preset points for Algorithm 2, and provide better initialization for the Algorithm 3.

$rand$ is generated. If $rand < Pr_e$, this path is taken as the current path; Otherwise, the original path is maintained. Then the current temperature drops and the iterative process continues,

$$T_e = T_e \times T_\alpha, \tag{31}$$

where T_α is the temperature attenuation coefficient. The coarse trajectory planning algorithm (CTPA) is summarized in Algorithm 2.

B. Automatic flying control algorithm (Algorithm3)

Due to the unknown jamming environment, the trajectory control for UAV is difficult to address by the traditional approaches, such as convex optimization theory which is widely used in the existing works. As mentioned early in Section II, the point-to-point precise trajectory control problem, i.e., $P2$, can be modeled as an MDP problem, with state space S, action space A, reward function R. Thus, a Q learning based algorithm is proposed to explore the unknown jamming environment and search for the optimal possible trajectory.

The Q function of the UAV is expressed as $Q(s_t, a_t)$, which is updated as follows:

$$\begin{aligned} Q_{t+1}(s_t, a_t) &= (1-\alpha) Q_t(s_t, a_t) \\ &+ \alpha \left(R(s_{t'}) + \gamma \max_{a_t'} Q_t(s_t', a_t') \right), \end{aligned} \tag{32}$$

where $\alpha = 1/(\omega(s_t, a_t) \lg(\omega(s_t, a_t)))$ is the learning rate [31],

$$\sum_{l=0}^{\infty} \alpha_l = \infty, \sum_{l=0}^{\infty} \alpha_l^2 < \infty, \tag{33}$$

where $\omega(s_t, a_t)$ is the times that the action a_t is selected in s_t. s_t' is the next state after performing action a_t, and a_t' is next possible action to perform. $\pi(a_t) \in [\pi(t, a_1), \pi(t, a_2), \ldots, \pi(t, a_4)]$ is the mixed strategy of the action selection, and $\pi(a_t)$ represents the probability to choose the action $a_t \in [a_1, a_2, a_3, a_4]$. Specially, $\pi(a_t)$ is given by:

$$\pi(a_t) = \frac{e^{Q_t(s_t, a_t)/\tau}}{\sum_{t'=1}^{4} e^{Q_t(s_t, a_{t'})/\tau}}, \tag{34}$$

where, τ is the Boltzmann model parameter,

$$\tau = \max\{\tau_0 e^{(-\upsilon t)}, \hat{\tau}\}, \tag{35}$$

τ_0 is related to the exploration time, and $\hat{\tau}$ is the ending condition in the exploration stage. υ affects the transition from exploration to exploitation.

To deal with the trajectory control problem studied in this paper, we have made modifications to the classical Q learning algorithm.

(1) Firstly, Q learning is generally used to deal with one-dimensional state and action space. Therefore, the position coordinates and state numbers of the UAV need to be converted from two-dimension to one-dimension. Moreover, after the action a_t is performed, the inverse transformation from one-dimension to two-dimension is needed to realize the state transfer $s_t \to s_t'$. Therefore, the state space, action performing,

Algorithm 2: Coarse trajectory planning algorithm (CTPA)

Input: The preset points set $l_p = [l_p^1, l_p^2, \ldots, l_p^N]$.
Output: The preliminary planning of rough trajectory $\xi^* = \Phi(l_p)$.

Initializes the location of the preset point and its weight matrix of propulsion energy consumption according to Eq. (13).
Initialize the annealing parameters including initial temperature T_{e0}, iteration number M_T, M_e, and temperature attenuation coefficient T_α.
Initializes a trajectory ξ_0, calculate the flying energy cost $\mathbb{E}_0^v$, $T_e = T_{e0}$.
for $m = 1$ **to** M_T **do**
  for $e = 1$ **to** M_e **do**
    Calculate the flying energy cost $\mathbb{E}_{m,e}^v$ of the current trajectory $\xi_{m,e}$ according to Eq. (19a).
    Obtain a new trajectory $\hat{\xi}_{m,e} = \Upsilon(\xi_{m,e})$.
    Calculate the flying energy cost $\hat{\mathbb{E}}_{m,e}^v$ of the new trajectory.
    if $\hat{\mathbb{E}}_{m,e}^v < \mathbb{E}_{m,e}^v$ **then**
      Update $\hat{\xi}_{m,e}$ as the current trajectory, $\xi_{m,e} \leftarrow \hat{\xi}_{m,e}$.
      else if $\hat{\mathbb{E}}_{m,e}^v \geq \mathbb{E}_{m,e}^v$ **then**
        Calculate Pr_e according to Eq. (30).
        if $rand < Pr_e$ **then**
          Update $\xi_{m,e} \leftarrow \hat{\xi}_{m,e}$.
        end
      end
    end
  end
  The temperature drops, $T_e = T_e \times T_\alpha$.
end

and state transition are all two-dimension.

(2) Secondly, in order to pay more attentions to the long-term reward of the whole path, we set a delayed return mechanism for the reward function. If agent successfully finds a path reaching to the destination grid, an additional reward is assigned to each pair of states and actions along that path.

(3) Thirdly, to reduce complexity, we adopt a block strategy. In the process of point-to-point trajectory control, we take out the square region with source grid and destination grid as vertices for pathfinding learning, which greatly reduces the computational complexity and improves the efficiency and effectiveness of the proposed scheme.

The Q learning based automatic flying control algorithm is summarized in Algorithm 3.

In addition, in order to deal with the challenge of the larger scale network, we propose another deep reinforcement learning (DRL) based ACFA algorithm. Although we have made improvements in Algorithm 3 to reduce the computational complexity, Q learning may not converge when the action-state space surges. In this case, we adopt DRL based scheme and utilized deep neural network instead of Q table to store learning experience, so as to effectively extract feature from

Algorithm 3: Automatic flying control algorithm (AFCA)

Input: The source location ξ_n, and the destination location ξ_{n+1}.
Output: UAV trajectory ψ_n from the preset point ξ_n to ξ_{n+1}, and the total uploading data $\mathbb{R}_n$.

Take out the square region with ξ_n and ξ_{n+1} as the vertices for pathfinding learning.
Initialize $Q_t(s_t, a_t)$=0, $\pi(a_t)$=1/4.
for $epoch = 1$ **to** K **do**
  $t = 1$, $s_t = \xi_n$, $\psi_n = [\xi_n]$.
  while $s_t \neq \xi_{n+1}$ & $t \leq L_{\xi_n,\xi_{n+1}}^{th}$ **do**
    In the current location s_t, UAV chooses action a_t from A according to $\pi(a_t)$, and gets into the next location $s_t{}'$ after the transformation of coordinates and dimensions. Updates $\psi_n = [\xi_n, \ldots, s_t, s_t{}']$.
    UAV uploads data traffic in the next location $s_t{}'$ and obtains the reward $R(s_t{}')$. Calculate the uploading data $\mathbb{R}_n \leftarrow \mathbb{R}_n + R(s_t{}')$.
    UAV updates $Q_{t+1}(s_t, a_t)$ and $\pi(a_t)$ according to Eq. (32) and (34).
    $t \leftarrow t + 1$, $s_t \leftarrow s_t{}'$.
  end
  If a feasible trajectory is found, each selected action on the trajectory will be given a delayed return R_0 to their Q value.
end

the complex environments and realize effective exploration of the large-scale network.

Based on our previous work in [31], we train the DRL network parameters according to proximal policy optimization (PPO) paradigm. The network structure and hyperparameters are set the same as [31]. The state action and reward function are the same as Algorithm 3, and we expanded the action space to the eight actions of "forward, backward, left, right, right forward, left forward, right backward, left backward".

DRL based approach can deal with the larger networks and longer paths, but it also has higher complexity. The DRL based automatic flying control approach is given in Algorithm 4.

C. Environment Situation Estimating Algorithm (Algorithm5)

An environment situation estimating algorithm is proposed to address $P3$, which is summarized in Algorithm 5. Based on the current selected preset points $l_p = [l_p^1, l_p^2, \ldots, l_p^N]$, the specific UAV flight trajectory can be obtained according to Algorithm 2 and Algorithm 3, and the data collected during the flight can be uploaded to the satellite. Based on the uploading data, the satellite can evaluate the situation information of the current environment according to Eq. (24) and (25). Based on the situation information and historical experience, satellite reselects the preset point l_p^* according to $\widetilde{\mathbb{D}}$ for larger traffic points, which is input to Algorithm 2. Meanwhile, it provides better initialized Q value according to $\widetilde{\mathbb{C}}$ for Algorithm 3. Finally, the satellite could get the final

Algorithm 4: DRL based automatic flying control algorithm (D-AFCA)

Input: The source location ξ_n, the destination location ξ_{n+1}, and DRL hyperparameters.
Output: The trained parameters by DRL model, UAV trajectory ψ_n from the preset point ξ_n to ξ_{n+1}, and the total uploading data $\mathbb{R}_n$.

Initialize hyperparameters for the DRL neural network.
for *epoch* $e = 1$ **to** E **do**
  Initialize the environment state s_0 as the source grid ξ_n.
  for *time step* $k = 1$ **to** $L^{th}_{\xi_n,\xi_{n+1}}$ **do**
    Obtain policy $\pi(s_k)$ using the DRL policy network.
    Sample the action a_k according to the ϵ-greedy policy of $\pi(s_k)$.
    Update the next state s_{t+1} with regard to $p(s_{k+1}|s_k, a_k)$.
    Estimate the instantaneous reward $r_k = R(s_k)$.
    Store the experience $\{s_k, s_{k+1}, r_k, a_k\}$.
    $k \leftarrow k + 1$.
  end
  Record the completed trajectory with the corresponding total uploading data.
  Update hyperparameters according to the PPO method.
  $e \leftarrow e + 1$.
end

Algorithm 5: Environment situation estimating algorithm (ESEA)

Input: The rough trajectory $\xi^* = \Phi(l_p)$ obtained by Algorithm 2, and the precise trajectory $\{\psi_n^*\}_{n=1}^{N-1}$ obtained by Algorithm 3, and the corresponding collected data.
Output: The assessed results of the hostile environment situation $\Theta = [\widetilde{\mathbb{D}}, \widetilde{\mathbb{C}}]$, and the final preset points set l_p^*.

Initializes the environment situation matrix $\Theta = 0$.
for *Each grid on the UAV's flight trajectory* l_t **do**
  Updates the average traffic flow and average beam quality of the current grid, $\widetilde{\mathbb{D}}$, $\widetilde{\mathbb{C}}$ according to Eq. (24) and (25).
  Reselect the preset data points set l_p^* based on the estimated situation to achieve the expected larger collected data, lower jamming impact, and better beam quality.
end

converged environment situation map, and the UAV can obtain the sub-optimal trajectory control scheme.

D. Complexity and Convergence Analysis

The proposed ADTO algorithm consists of three sub-algorithms: the CTPA algorithm (Algorithm 2), the AFCA algorithm (Algorithm 3), and the ESEA (Algorithm 5). Firstly, Algorithm 2 provides a rough trajectory which including all of the preset points only. Then, Algorithm 3 realizes the automatically flying trajectory control between two preset points of the rough trajectory obtained by Algorithm 2. Finally, Algorithm 5 estimates the environment situation by the uploaded data which is collected during the flight, and adjusts the preset points which are used as the new input to Algorithm 2. Meanwhile, the environment situation results can provide better initialized Q value for Algorithm 3.

1) The computational complexity of the ADTO algorithm: As for each flight, the computational complexity $\mathcal{O}_1$ of the proposed ADTO algorithm mainly focuses on the three sub-algorithm (Algorithm 2, 3, 5), i.e., $\mathcal{O}_1 = (\mathcal{O}_2 + \mathcal{O}_3 + \mathcal{O}_5)$. The scalar multiplication cost of Algorithm 2 is mainly focus on Step 6, 8, 12, thus, $\mathcal{O}_2 = \mathcal{O}\left(M_T(1 + M_e(2N-1))\right)$, where $2(N-1)$ is the computational cost of the Step 6 and 8, and N is the cardinality of ξ, i.e., $N = |\xi|$. The scalar multiplication cost of Algorithm 3 mainly focuses on the Step 7, whose scalar multiplication cost is $\mathcal{O}_3 = \mathcal{O}\left(KL^{th}\lfloor T_l/T_b \rfloor\right)$, where K is the iteration number, L^{th} is the maximum pathfinding length, and $\lfloor T_l/T_b \rfloor \leq \sqrt{2}R_B/v_u/T_b$ is the beam switching times and it is the scalar multiplication cost of $\hat{D}_l$. The scalar multiplication cost of Algorithm 4 is $\mathcal{O}_5 = \mathcal{O}\left(2\Pi \sum_{n=1}^{N-1} |\psi_n|\right)$, where Π is the scalar multiplication cost of operation of segment averaging E, and $|\psi_n|$ is the cardinality of the n-th precise trajectory.

2) The convergence analysis of the ADTO algorithm: Similarly, due to the finite iteration times, the convergence of the proposed ADTO algorithm also depends on the convergence of the three sub-algorithms.

As for the Algorithm 2, if a new better trajectory appears, it will be updated as the current trajectory. Then, the utility will be improved according to the Pareto criterion. Due to the limited trajectory choice and finite return utility, the final trajectory and utility will converge to the stable value. Meanwhile, as the iteration increases, the temperature decreases and the acceptance probability Pr_e decreases as well. After enough iterations, the very low acceptance probability keeps the current trajectory unchanged.

As for the Algorithm 3, the learning process could get stable due to the fixed source point ξ_n and destination points ξ_{n+1} and the limited flying distance $L^{th} < \infty$. According to the accumulated experience during the learning process, the UAV will choose the probabilistic sub-optimal trajectory to maximize the expected uploading data. The convergence of this Q-Learning based trajectory control algorithm can be proved. Based on [43], if Eq. (33) is met, the proposed Q-Learning based algorithm can converge to a stable point.

As for the Algorithm 5, the size of the grid network is limited, so is the collected data from each flight. Algorithm 4 is an evaluation algorithm, which is mainly based on the analysis of collected data. Therefore, the specific evaluation conclusions can be obtained under the premise of limited data sources. However, the assessment results tends to be more accurate as the increasing of the collected data. This trend will be accomplished in finite iterations, because the hostile

TABLE I: The parameters of Shadowed-Rician model and Jamming link model.

Shadowed-Rician model	χ	m	μ	**Nakagami-m model**	θ	η
Heavy shadowing model	0.063	0.739	8.97×10^{-4}	Heavy jamming	4	2
Average shadowing model	0.126	10.1	0.835	Average jamming	3	1
Light shadowing model	0.79	97	6.45	Light jamming	1	1

TABLE II: Simulation parameters.

Parameters	**Value**
Earth radius	$R_E = 6371$km
Minimum elevation angle	$\vartheta = 10^\circ$
Orbit height	$H_s = 500$km
Constellation parameters	$N_o = 10, N_s = 21$
Beam square number	$N_B = 12$
Length of each beam square	$R_B = 50.89$km
Beaming switch distance	$R_S = 29.81$km
Coverage time	$T_c = 6.41$min
Beam switching time	$T_b = 0.53$min
Mean traffic in hot grids	$\lambda^h \in [550, 1100]$Mbit
Mean traffic in others	$\lambda^l \in [50, 100]$Mbit
UAV flying height	$H_u = 10$km
UAV flying speed	$v_u = 15$km/min
Channel noise	$\sigma = 0.1$W
Channel bandwidth	$B = 2$MHz
Maximum sub-satellite half angle	$\phi = 63.79^\circ$
Geocentric angle	$\alpha = 12.21^\circ$
UAV transmission power	$p_U = 1$W
SJNR threshold	$r_{th} = 0.1$
Jamming power	$p_J = 3$W
Jammer probability	$Pr_J = 0.9$
Path loss exponent	$\beta = 2$
Flying energy coefficient	$e_1 = 0.1$W
Uploading energy coefficient	$e_2 = 0.01$W/Mbit
Cache stack capacity	$\lvert\mathcal{K}\rvert = 1500$Mbit
Flight distance threshold	$L^{th} = 15$
SA parameters Te_0, T_α	$1000, 0.95$
RL parameters $\tau_0, \hat{\tau}, \gamma$	$10^7, 0.1, 0.6$

environment is unknown but not unknowable.

In summary, the proposed ADTO algorithm consists three sub-algorithms, and the whole algorithm framework runs iteratively according to Pareto criterion. According to the limited utility and Pareto principle, the proposed ADTO algorithm is convergent and stable.

3) The optimality analysis of the ADTO algorithm: Based on Remark 1 and Remark 2, due to the environment setting, the proposed closed-loop three-step approach can converge to a sub-optimal solution. As the number of flights increases, the UAV will have more accumulated knowledge of the environment, and the optimized trajectory will gradually move towards the approximately optimal solution.

IV. Simulation Results

In this section, simulation is carried out to validate the effectiveness of the proposed approach with MATLAB 2016a and Python 3.9.6. Each grid has three shadowing model options and three jamming link status which are shown in Table I. Other parameters are shown in Table II.

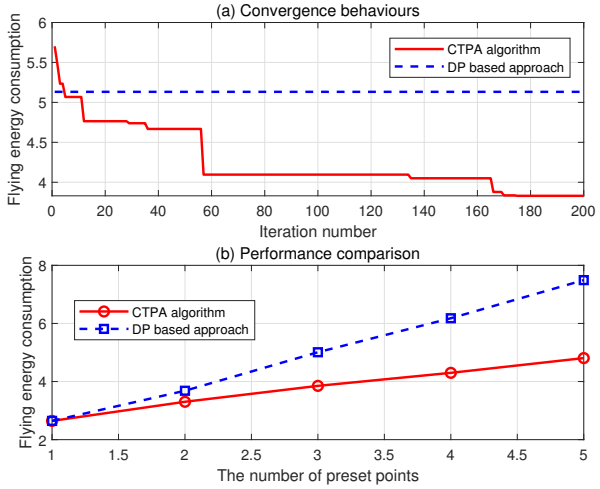


Fig. 5: The convergence behaviours of the proposed CTPA algorithm and DP based algorithm.

Since the TSP problem is a NP hard problem, there is no effective optimal solution. Dynamic programming (DP) algorithm is an efficient traditional algorithm to solve the TSP problem. We compare the proposed CTPA approach with the DP based algorithm. As shown in Fig. 5, the proposed CTPA algorithm continuously improves performance as the number of iterations, and the energy consumption is significantly reduced. However, the DP based algorithm not only requires global information, but also dramatically increases the energy consumption and complexity due to the increasing points. In addition, as the increasing number of the preset points, the proposed CTPA algorithm has lower energy consumption than DP based algorithm.

ϵ greedy based RL algorithm for anti-jamming trajectory control and the proposed AFCA algorithm are compared in this section. As indicated in Fig. 6, the proposed AFCA algorithm has better performance in utility and convergence. The automatically precise control trajectory between two preset points and the corresponding section assessment results obtained by this learning process are shown in Fig. 7. We can see that the environment dynamic is almost obtained by UAV during the automatic learning process, and it automatically makes the probabilistically sub-optimal trajectory control with a better channel state to reach the destination.

In Fig. 8, we compare the Q learning based approach, DRL based approach and the mean theoretical value. We adopt the PPO paradigm in the DRL based approach, and

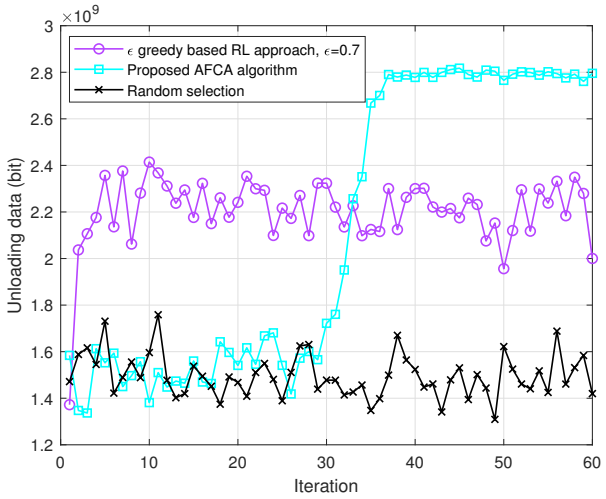


Fig. 6: The comparison of different solving approaches in point to point flight.

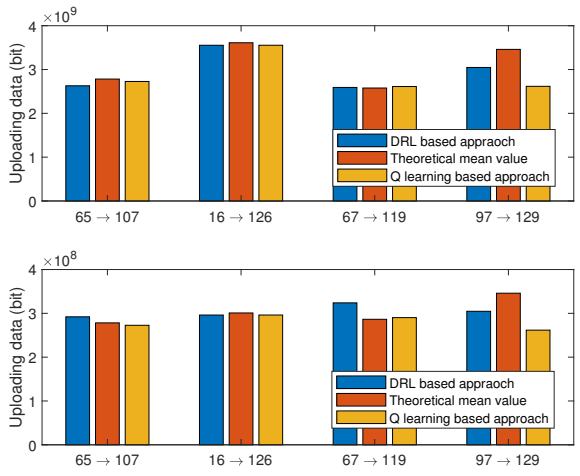


Fig. 8: The comparison between Q learning based AFCA approach and DRL based AFCA approach.

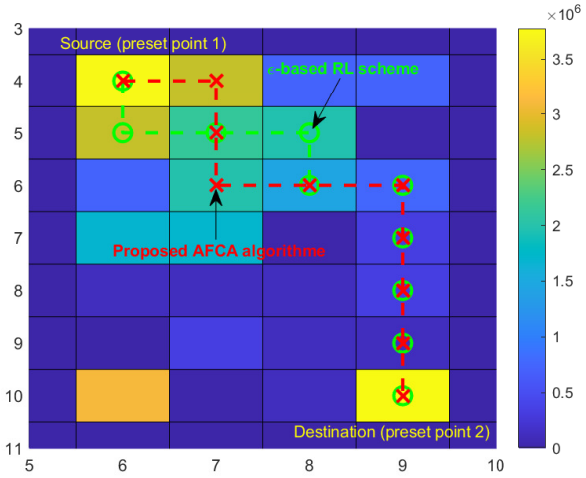


Fig. 7: The partly environmental information obtained during this point to point flight, and the corresponding flight trajectory.

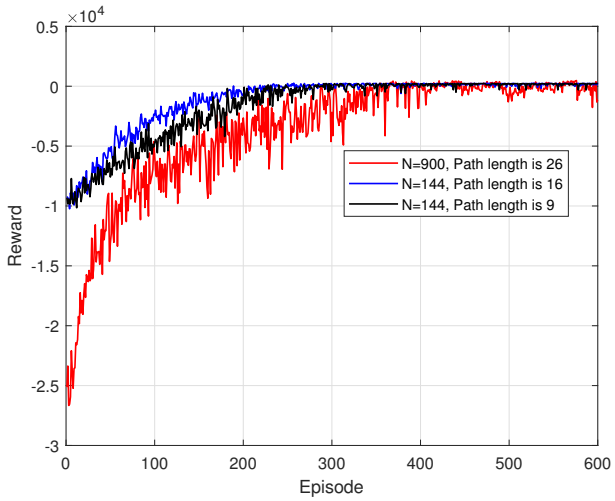


Fig. 9: The influence of grid number on DRL based AFCA algorithm.

the mean theoretical value is obtained by the exhaustively-search-then-select-the-best method. As shown in Fig. 8, when the network size is small, the Q learning based algorithm and DRL based algorithm have similar performance, and they are all below or equal to the theoretical best value due to that the path length is short and the action-state space is limited. In addition, compared with DRL based approach, Q learning based approach has lower computational complexity.

However, when the network size increases and the path length becomes longer, it is difficult for Q learning algorithm to get a convergent effective solution. As shown in Fig. 9, DRL based approach can still get a convergent solution, and the convergence speed is weakly correlated with the network size. Because DRL algorithm uses deep neural network instead of Q table to store the experience information, it can extract the features of complex environment more effectively.

The collected data obtained during the UAV flight is uploaded to the satellite. The satellite assessment process of environment situation based on the data analysis are shown in Fig. 10. Specifically, Fig. 10(a)-(c) show the intermediate assessment results of channel state, and (d)-(f) are the assessment results of traffic distribution, which could be used to further guide the next flight in the current hostile environment.

The normalized estimation deviation during the assessing process is shown in Fig. 11, where each trajectory optimization not only aims to better complete the current task, but also undertakes the task of exploring the environment and obtaining more comprehensive and accurate environmental information.

Fig. 12 compare the uploading data of our proposed ADTO algorithm and other compared schemes. It is indicated that the proposed ADTO algorithm can achieve higher uploading data than the centralized design schemes. In fact, this trajectory is not only highly coincident with the hot points with high traffic, but also seeks to avoid areas with severe jamming and poor beam quality to pursue higher transmission efficiency. Due to

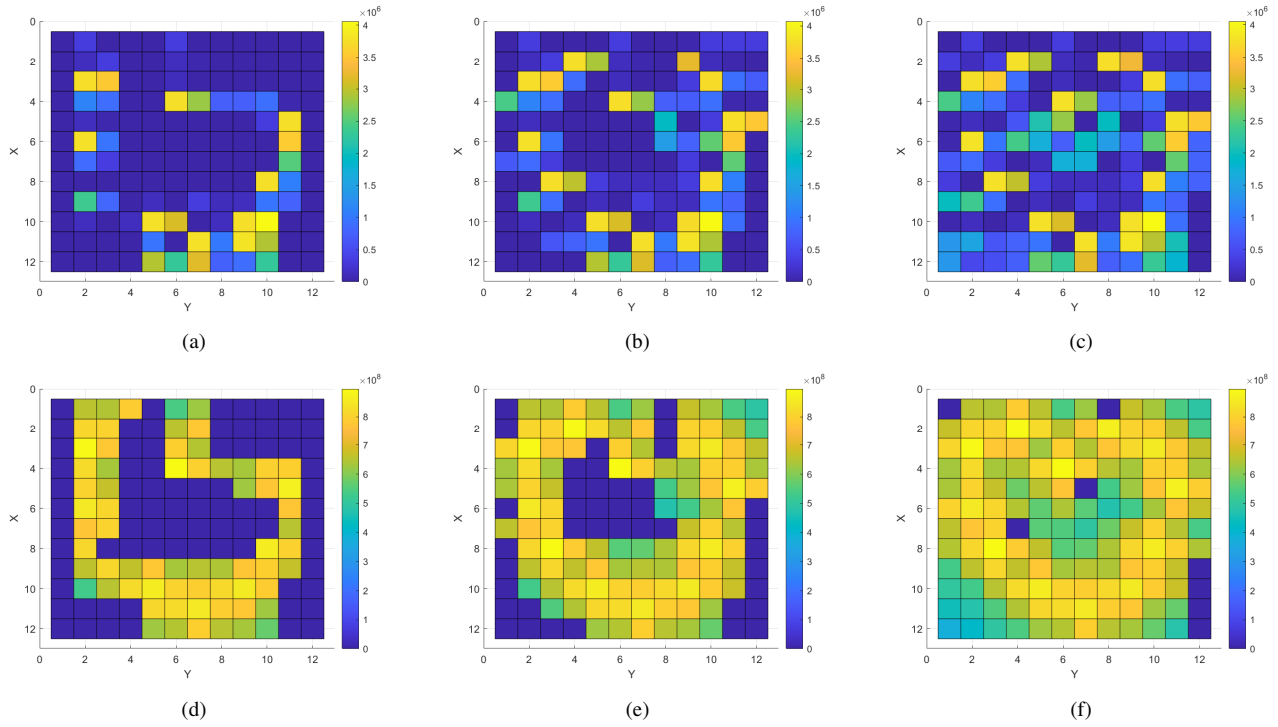


Fig. 10: The assessment process of environment situation. (a)-(c) show the intermediate assessment results of channel state, and (d)-(f) are the assessment results of traffic distribution. Although there are certain grids that have not been reached, they are all non-hot grids. Hot grids are easy to spot due to their aggregation. Moreover, due to the possibility of random exploration in Algorithms 2 and 3, the UAV will eventually pass through all the grids after enough flights.

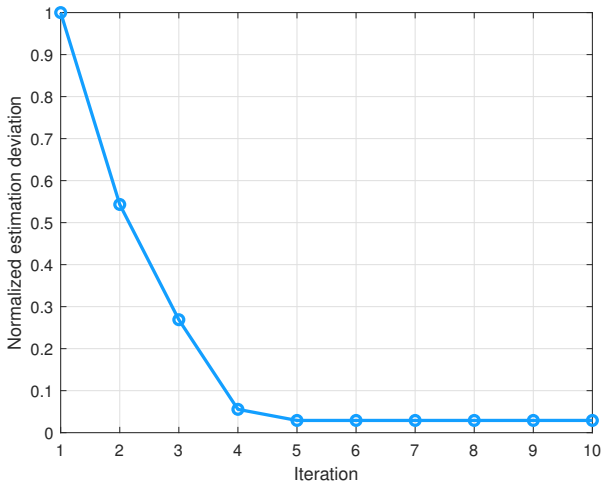


Fig. 11: The normalized estimation deviation during the trajectory planning process obtained by the ESEA algorithm.

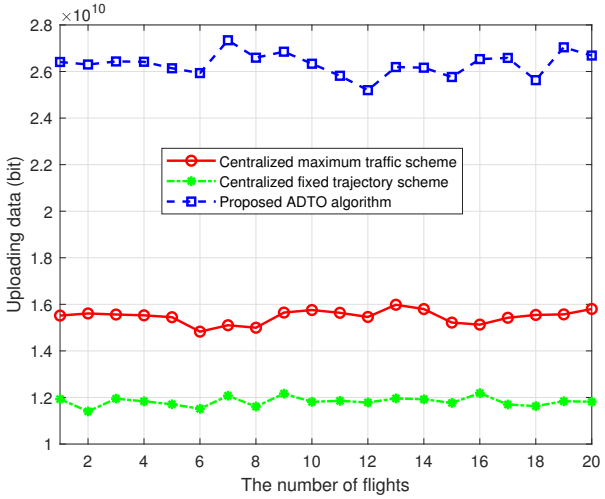


Fig. 12: The comparison on uploading data of different trajectory design schemes.

the uncertainty of the hostile environment, the channel state of each grid is difficult to obtain, thus the centralized optimization scheme can only carry out the beforehand optimization of UAV trajectory to maximize the collected data, or make a fixed flight trajectory according to the satellite observation information.

The influence of jamming probability Pr_d and flying distance threshold L^{th} are indicated in Fig. 13. If the UAV is provided with more energy reserves to support flight, it can fly autonomously over a wider area, which is conducive to the reconnaissance analysis of the environmental situation. As the L^{th} increases, the estimation error decreases gradually. Meanwhile, the increase of jamming probability will seriously affect the autonomous flight and data collection, which further reduces the estimation accuracy of environmental situation. Therefore, larger energy reserve and lower jamming probability are beneficial to improve environmental situational

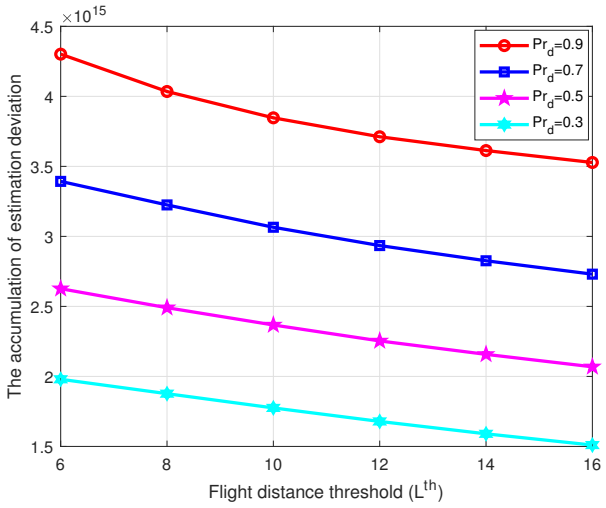


Fig. 13: The influence of jamming probability and flight distance threshold.

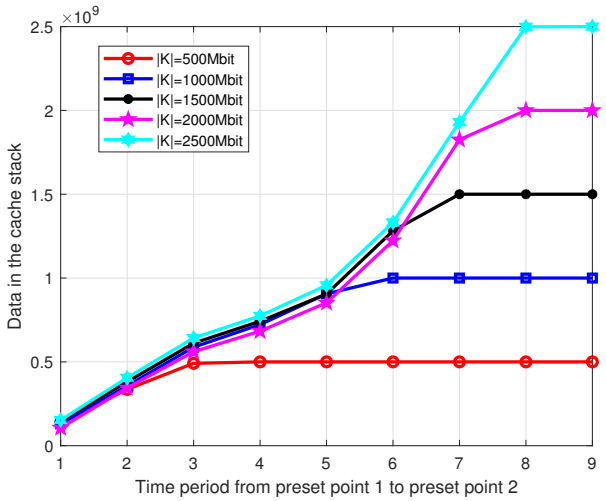


Fig. 14: The data changes in the cache stake with different cache capacity which is determined by the energy reserve.

awareness.

As shown in Fig. 14, the cache stack capacity of UAV has an important impact on data collection task. If the data to be collected in the current environment is significantly greater than the data transmission rate, the cache stack has to abandon a lot of data due to the capacity limit. Therefore, sufficient cache stack capacity is of great significance for the effectiveness and real-time performance of data collection task in hostile jamming environment.

V. Conclusion

This paper investigated the satellite-assisted UAV trajectory control problem in the hostile jamming environments. Due to the uncertain hostile environment, the anti-jamming trajectory control approach was proposed based on the limited and partial information, and decoupled as a closed-loop three-step approach: preliminary planning of coarse trajectory, point-to-point precise trajectory control and environmental situation assessment. The simulation results proved that the proposed algorithm can finally find the sub-optimal trajectory, and the UAV can complete the data collection of hot points and upload more data during the flight. Based on these data, the satellite can get the available situation map of the hostile environments.

References

[1] C. Han, A. Liu, X. Liang, L. Ruan, and K. Cheng, "Uav trajectory control against hostile jamming in satellite-uav coordination networks," in *Proc. IEEE ICCC*, 2020, pp. 701–705.

[2] F. Dario and W. R. J, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, no. 7553, pp. 460–466, 2015.

[3] O. Kodheli, E. Lagunas, N. Maturo, S. K. Sharma, B. Shankar, J. F. M. Montoya, J. C. M. Duncan, D. Spano, S. Chatzinotas, S. Kisseleff, J. Querol, L. Lei, T. X. Vu, and G. Goussetis, "Satellite communications in the new space era: A survey and future challenges," *IEEE Commun. Surv. Tutor.*, pp. 1–1, 2020.

[4] N. C. Coops, T. R. H. Goodbody, and L. Cao, "Four steps to extend drone use in research," *Nature*, vol. 572, no. 7770, pp. 433–435, 2019.

[5] S. Zhang and J. Liu, "Analysis and optimization of multiple unmanned aerial vehicle-assisted communications in post-disaster areas," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12 049–12 060, 2018.

[6] M. Mozaffari, A. Taleb Zadeh Kasgari, W. Saad, M. Bennis, and M. Debbah, "Beyond 5g with uavs: Foundations of a 3d wireless cellular network," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 357–372, 2019.

[7] C. Han, A. Liu, L. Huo, H. Wang, and X. Liang, "A prediction-based resource matching scheme for rentable leo satellite communication network," *IEEE Commun. Lett.*, vol. 24, no. 2, pp. 414–417, 2020.

[8] C. Liu, W. Feng, Y. Chen, C. Wang, and N. Ge, "Cell-free satellite-uav networks for 6g wide-area internet of things," *IEEE J. Sel. Areas Commun.*, pp. 1–1, 2020.

[9] S. Gu, Y. Wang, N. Wang, and W. Wu, "Intelligent optimization of availability and communication cost in satellite-uav mobile edge caching system with fault-tolerant codes," *IEEE Trans. Cogn. Commun. Netw.*, pp. 1–1, 2020.

[10] X. Zhang, W. Cheng, and H. Zhang, "Heterogeneous statistical qos provisioning over airborne mobile wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2139–2152, 2018.

[11] T. Qi, W. Feng, and Y. Wang, "Outage performance of non-orthogonal multiple access based unmanned aerial vehicles satellite networks," *China Commun.*, vol. 15, no. 5, pp. 1–8, 2018.

[12] M. Vondra, M. Ozger, D. Schupke, and C. Cavdar, "Integration of satellite and aerial communications for heterogeneous flying vehicles," *IEEE Netw.*, vol. 32, no. 5, pp. 62–69, 2018.

[13] J. Lyu, Y. Zeng, and R. Zhang, "Uav-aided offloading for cellular hotspot," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3988–4001, 2018.

[14] J. Zhang, Y. Zeng, and R. Zhang, "Uav-enabled radio access network: Multi-mode communication and trajectory design," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5269–5284, 2018.

[15] D. H. Tran, T. X. Vu, S. Chatzinotas, S. ShahbazPanahi, and B. Ottersten, "Coarse trajectory design for energy minimization in uav-enabled," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9483–9496, 2020.

[16] O. Esrafilian, R. Gangula, and D. Gesbert, "Learning to communicate in uav-aided wireless networks: Map-based approaches," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1791–1802, 2019.

[17] F. Cheng, S. Zhang, Z. Li, Y. Chen, N. Zhao, F. R. Yu, and V. C. M. Leung, "Uav trajectory optimization for data offloading at the edge of multiple cells," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6732–6736, 2018.

[18] J. Xu, Y. Zeng, and R. Zhang, "Uav-enabled wireless power transfer: Trajectory design and energy optimization," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5092–5106, 2018.

[19] O. M. Bushnaq, A. Celik, H. Elsawy, M. Alouini, and T. Y. Al-Naffouri, "Aeronautical data aggregation and field estimation in iot networks: Hovering and traveling time dilemma of uavs," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4620–4635, 2019.

[20] Y. Zeng and R. Zhang, "Energy-efficient uav communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, 2017.

[21] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-uav enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, 2018.

[22] X. Li, W. Feng, Y. Chen, C. Wang, and N. Ge, "Maritime coverage enhancement using uavs coordinated with hybrid satellite-terrestrial networks," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2355–2369, 2020.

[23] M. Hua, Y. Wang, M. Lin, C. Li, Y. Huang, and L. Yang, "Joint comp transmission for uav-aided cognitive satellite terrestrial networks," *IEEE Access*, vol. 7, pp. 14 959–14 968, 2019.

[24] A. Mostaani, O. Simeone, S. Chatzinotas, and B. Ottersten, "Learning-based physical layer communications for multiagent collaboration," in *Proc. PIMRC*, 2019, pp. 1–6.

[25] A. Mostaani, T. X. Vu, S. Chatzinotas, and B. Ottersten, "State aggregation for multiagent communication over rate-limited channels," in *Proc. GLOBECOM*, 2020, pp. 1–7.

[26] B. Wang, Y. Wu, K. R. Liu, and T. C. Clancy, "An anti-jamming stochastic game for cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 877–889, 2011.

[27] Y. Yuan, L. Lei, T. X. Vu, S. Chatzinotas, and B. Ottersten, "Actor-critic deep reinforcement learning for energy minimization in uav-aided networks," in *Proc. EuCNC*, 2020, pp. 348–352.

[28] Y. Yuan, L. Lei, T. X. Vu, S. Chatzinotas, S. Sun, and B. Ottersten, "Actor-critic learning based energy optimization for uav access-and-backhaul networks," *EURASIP J. Wirel. Commun. Netw.*, pp. 1–1, 2021.

[29] C. Han and Y. Niu, "Multi-regional anti-jamming communication scheme based on transfer learning and q learning." *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 7, 2019.

[30] C. Han, A. Liu, H. Wang, L. Huo, and X. Liang, "Dynamic anti-jamming coalition for satellite-enabled army iot: A distributed game approach," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 10 932–10 944, 2020.

[31] C. Han, L. Huo, X. Tong, H. Wang, and X. Liu, "Spatial anti-jamming scheme for internet of satellites based on the deep reinforcement learning and stackelberg game," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5331–5342, 2020.

[32] C. Han and Y. Niu, "Cross-layer anti-jamming scheme: A hierarchical learning approach," *IEEE Access*, vol. 6, pp. 34 874–34 883, 2018.

[33] J. Cho, J. Sung, J. Yoon, and H. Lee, "Towards persistent surveillance and reconnaissance using a connected swarm of multiple uavs," *IEEE Access*, vol. 8, pp. 157 906–157 917, 2020.

[34] L. Ruan, J. Wang, J. Chen, Y. Xu, Y. Yang, H. Jiang, Y. Zhang, and Y. Xu, "Energy-efficient multi-uav coverage deployment in uav networks: A game-theoretic framework," *China Commun.*, vol. 15, no. 10, pp. 194–209, 2018.

[35] I. D. Moscholios, V. G. Vassilakis, P. G. Sarigiannidis, N. C. Sagias, and M. D. Logothetis, "An analytical framework in leo mobile satellite systems servicing batched poisson traffic," *IET Commun.*, vol. 12, no. 1, pp. 18–25, 2017.

[36] K. An, M. Lin, W. Zhu, Y. Huang, and G. Zheng, "Outage performance of cognitive hybrid satellite–terrestrial networks with interference constraint," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 9397–9404, 2016.

[37] K. An, M. Lin, and T. Liang, "On the performance of multiuser hybrid satellite-terrestrial relay networks with opportunistic scheduling," *IEEE Commun. Lett.*, vol. 19, no. 10, pp. 1722–1725, 2015.

[38] Z. Gao, A. Liu, C. Han, and X. Liang, "Max completion time optimization for internet of things in leo satellite-terrestrial integrated networks," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9981–9994, 2021.

[39] L. You, K. X. Li, J. Wang, X. Gao, X. G. Xia, and B. Ottersten, "Massive mimo transmission for leo satellite communications," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1851–1865, 2020.

[40] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing uav," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, 2019.

[41] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, 1997.

[42] C. Groba, A. Sartal, and X. H. Vázquez, "Solving the dynamic traveling salesman problem using a genetic algorithm with trajectory prediction: An application to fish aggregating devices," *Comput. Oper. Res.*, vol. 56, pp. 22–32, 2015.

[43] A. Kianercy and A. Galstyan, "Dynamics of boltzmann q learning in two-player two-action games," *Phys. Rev. E*, vol. 85, no. 4, pp. 1574–1604, 2012.