Trajectory Optimization for UAV Emergency Communication with Limited User Equipment Energy: A safe-DQN Approach

Tiankui Zhang, *Senior Member, IEEE,* Jiayi Lei, Yuanwei Liu, *Senior Member, IEEE,* Chunyan Feng, *Senior Member, IEEE* and Arumugam Nallanathan, *Fellow, IEEE*

(*Invited Paper*)

Abstract

In post-disaster scenarios, it is challenging to provide reliable and flexible emergency communications, especially when the mobile infrastructure is seriously damaged. This article investigates the unmanned aerial vehicle (UAV)-based emergency communication networks, in which UAV is used as the mobile aerial base station for collecting information from ground users in affected areas. Due to the breakdown of ground power system after disasters, the available energy of affected user equipment (UE) is limited. Meanwhile, with the complex geographical conditions after disasters, there are obstacles affecting the flight of UAV. Aiming at maximizing the uplink throughput of UAV networks during the flying time, we formulate the UAV trajectory optimization problem considering UE energy limitation and location of obstacles on the ground. Since the constraint on UE energy is dynamic and long-term cumulative, it is hard to be solved directly. We transform the problem into a constrained Markov decision-making process (CMDP) with UAV as agent. To tackle the CMDP, we propose a safe-deep-Q-network (safe-DQN) based UAV trajectory design algorithm, where the UAV learns to selects the optimal action in reasonable policy sets. Simulation results reveal that: i) the uplink throughput of the proposed algorithm converges within multiple iterations; and ii) compared with the benchmark algorithms, the proposed algorithm performs better in terms of uplink throughput and UE energy efficiency, achieving a good trade-off between UE energy consumption and uplink throughput.

This work was supported by the National Key R&D Program of China under Grant 2019YFC1511302 and the National Natural Science Foundation of China under Grant 61971060.

Tiankui Zhang, Jiayi Lei and Chunyan Feng are with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: {zhangtiankui, leijiayi, cyfeng}@bupt.edu.cn).

Yuanwei Liu and Arumugam Nallanathan are with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K. (e-mail: {yuanwei.liu, a.nallanathan}@qmul.ac.uk).

Index Terms

constrained Markov decision-making process, emergency communication, trajectory design, deep reinforcement learning

I. Introduction

Large-scale natural disasters always inflict severe and unpredictable loss of life and property. In the past 30 years, various types of natural disasters, such as earthquakes, tsunamis, floods, wildfires, hurricanes, etc., have resulted in many deaths, and material losses caused by disasters worldwide have increased by approximately 100%-150% [1]. When a disaster occurs, maintaining real-time communications help to obtain post-disaster situational awareness, which can greatly improve the efficiency of rescue missions. Unfortunately, in most cases, disasters will damage the communication equipment, making the communication network, which nowadays predominantly depends on wireless communication infrastructure, unable to function normally. During the hurricane Harvey in the U.S., the FCC published that only one of the 19 cell towers in Aransas County in Texas was functioning and 85 percent of cellular towers became offline in nearby Counties [2]. Therefore, it is very necessary to establish emergency communications with rapid response and flexible networking.

Considering the complex ground conditions and the lack of power supply during post-disaster, the emergency communication network should be highly energy efficient, simple deployment, and have good compatibility among different user devices and different types of disasters [2]. Among numerous emergency communication networking technologies, it’s an efficient and feasible solution to deploy unmanned aerial vehicles (UAVs) with flexible deployment and timely response as the mobile aerial BS to construct a mobile emergency communication network [3]. Currently, UAV has been widely used in different disaster management applications, including monitoring and early warnings, disaster information fusion and sharing, supply dropping, damage assessment and so on. What’s more, as the movable characteristic of UAV allows the distance between the receiver and the transmitter to be adjusted in real time, which helps to deal with the problem of low UE signal level in post-disaster scenarios, UAV BS can be used as an important communication facility to build a standalone communication system in post-disaster areas [4].

Although the UAV emergency communication networks play a powerful role in a disaster scenario, there are still some key technical difficulties: 1) the working time of UAV is limited by on-broad battery of UAV [5]; 2) the trajectory plan of UAV requires timely and accurate response to emergencies in complex and harsh geographical environment of natural disasters filed [3]; 3) In addition, the available energy to equipment of trapped users is also extremely limited due to the damage to the crucial infrastructures (such as power supply) [6]. Based on the above considerations, the UAV emergency communications should be completed as far as possible before the user's equipment runs out of energy within the working time of UAV.

A. Motivations and Related Works

Due to the high flexible mobility, UAV has attracted significant research interest in the field of wireless communication [7]. There are many researches that combine UAV with different communication technologies, such as non-orthogonal multiple access [8–10], massive MIMO [11], millimeter wave communication [12] and reconfigurable intelligent surfaces [13]. Meanwhile, caching-enabled UAV cellular networks has attracted increasing attention to effectively alleviate the traffic load of wireless backhaul links [14, 15]. UAV can also be used as the mobile relay to provide a new access method for resource constrained users, thus increasing the throughput of the whole system [16]. In addition, UAV has been also applied in various specific scenarios [17–20]. Zhang et al. [17] studied the content distribution in hot areas, and proposed the cache-enabling UAV-assister cellular network which successfully improved the quality of user experience (QoE). In [18], UAV acts as a MEC server and provides communication and computing services for terminal devices in the Internet of things. In [19] and [20], UAVs are used to provide wireless energy harvesting and information transmission for ground users. On the other hand, with the rapid development of artificial intelligence technology, the application of reinforcement learning (RL) in wireless communication network has become a research hotspot [21]. Some researchers have applied RL to UAV networks to make the UAV wireless communication more efficient and adaptable [22–25]. Yin et al. [22] studied the trajectory design in UAV-assisted cellular network. The optimization problem for maximizing the uplink transmission rate was transformed into a Markov decision process, which was solved by deterministic policy gradient (DPG) algorithm. A long-term resource allocation problem in multi-UAV communication networks was formulated as a stochastic game for maximizing the expected rewards in [23], which was solved by a multi-agent reinforcement learning framework. In [24], with the goal of maximizing the energy efficiency and coverage of UAV communication network, an actor-critic based deep enhancement learning algorithm was used to optimize the flight direction and flight distance of the UAV. Based on the prediction of user's mobility, Liu et al. [25] proposed a multi-agent Q-learning-based trajectory design and power control algorithm to maximize the transmission rate in multi-UAV assisted wireless networks.

Although excellent research has been conducted on UAV communications, there are few works focusing on UAV-assisted emergency communication networks in disasters [26–29]. Merwaday et al. [26] used a genetic algorithm to get the best location of the UAV, thereby improving the network throughput.The problem that maximizing the number of service users under limited UAV battery capacity by optimizing the flight path was proposed in [27]. This optimization task was transformed into a multi-armed bandit problem, and distance-aware upper confidence bound algorithm (D-CUB) and ε-exploration algorithm were proposed to solve it. Some encouraging work was done by Zhao et al. to establish a framework for UAV-assisted emergency networks in disasters [28]. There are three different network models corresponding to three scenarios: First, UAV is deployed to assist the surviving BSs; Second, when all ground BSs are destroyed, UAV serves as a flying base station to provide communication services; In addition, hovering UAVs are used as multi-hop relays to exchange the information between the disaster area and outside. The collection and transmission of user information in emergency scenarios considering natural environment and UAV energy consumption constraints were investigated in [29]. In order to improve the QoE and shorten the flight time of UAV, a path optimization scheme including hover point selection and mobility planning is proposed and solved by convex optimization method.

These existing works related to UAV-based emergency communication networks mainly pay attention to the energy consumption of UAV, but ignore the limitation on energy of ground user equipment (UE) caused by the paralysis of ground power transmission system and constrained user mobility after disasters. Meanwhile, most of researches assume that the UAV trajectory or deployment position at a certain altitude is not restricted by geographical conditions. However, as obstacles that are far above the ground such as residential buildings, office buildings and mountains are inevitably distributed, it is often difficult to find an airspace where UAVs can move freely in most practical scenarios. These obstacles will affect the flight of UAV and cause possible collisions in pratical application. Different from the existing works, we proposed a UAV-based emergency communication network, in which the energy limitation of UE is considered. In addition, we also notice the influence of air obstacles on UAV flight path. Thus, our proposed framework further enhances the feasibility of UAV emergency communication system, as compared with the existing works.

B. Contributions and Organization

As mentioned above, the emergency communication scenarios of current studies rarely consider the constrains on energy of UEs and obstacles in post-disaster areas. To fulfill this gap, a UAV-based emergency communication network with limited UE energy is researched in this article, in which the UAV acts as a mobile aerial BS to complete bits transmit from devices of users in affected area. The data collection task during disasters is always extremely urgent, however the coverage of UAV is relatively small. When the uploaded data of ground UEs is limited, the UAV trajectory need to be planed reasonably to increase the UEs' access opportunities, so as to collect as much user information as possible during the flight time. Therefore, our goal is to maximize the long-term uplink throughput of the system during the flying time by designing the flight trajectory of UAV. The main contributions are summarized as follows:

- We propose a framework of UAV-based emergency communication networks to collect user information in post-disaster areas. The terrestrial devices within coverage of the UAV can access to the mobile aerial BS when other mobile infrastructures are out of services. Considering the limitation on geographical conditions and energy supply in reality, we formulate a dynamic long-term optimization problem to maximize uplink throughput of UAV network during the flying time by optimizing UAV trajectory.
- We transform the original problem to a constrained Markov decision process (CMDP) with UAV as agent, in which the action, reward, and cost are defined as flight direction, uplink throughput and energy consumption of UE respectively. For the long-term cumulative constraint on energy consumption of UE, we first obtain a set of safe policies by constructing a reasonable Lyapunov function, and then we propose a safe-DQN based algorithm to solve the optimal policy in the safe set. For the constraint on avoiding obstacles, we define the concept of legal actions to tackle it.
- We demonstrate the feasibility and effectiveness of the proposed algorithm by numerical simulations. Simulation results show that the proposed UAV trajectory design algorithm converges after multiple iterations. Compared with benchmark algorithms, the proposed algorithm is able to effectively avoid collision during the UAV flight and gets a trade-off between system throughput and energy consumption of UEs. Besides, we also investigate the influence of UAV height by simulation.

The rest of this article is organized as follows. Section II presents the system model and formulates the optimization problem for long-term uplink throughput maximization. In Section III, we transform the problem into a CMDP and propose the safe-DQN based algorithm for trajectory design. Simulation results are provided in Section IV, and finally we conclude this paper in Section V.

II. System Model and Problem Formulation

Consider a post disaster rescue scenario with aerial obstacles, such as mountains or buildings, where rescuers can not approach easily. Due to the destruction of external forces (such as earthquake, flood, war, etc.), the ground infrastructure communication facilities in the certain area can’ t work normally. Furthermore, due to the destruction of infrastructure, the UE signal that can be received is often weak in disaster areas. In this case, the UAV can be used as a mobile aerial BS to establish temporary communication connection, and provide assistance for rescue by efficiently collecting information from affected users, as shown in Fig.1. We assume that there are K users trapped in the area, denoted by $\mathcal{K} = \{1, ...K\}$, and the corresponding locations are represented by $l_k \in \mathbb{R}^{2\times1}, k \in \mathcal{K}$. Taking into account the limited endurance of UAV, we assume that the continuous working time of the UAV is T. The UAV takes off from the fixed starting point and flies over the area along a specific trajectory at a constant speed v. When the time is up, the UAV lands back to the starting point to charge or replace its battery.

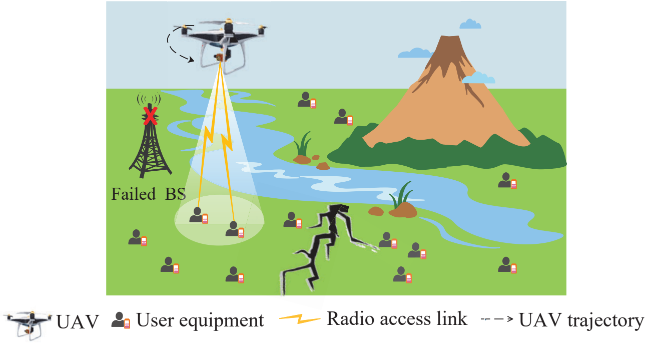


Fig. 1: UAV emergency communication networks.

A. UAV Mobility Model

For the convenience of illustration, we divide the UAV working duration T into M equal time slots with length δ_t, i.e.$T = M\delta_t$. Note that the value of δ_t is small enough to satisfy $\delta_t v \ll H$, where H is the flying height of UAV. So that in a time slot, the UAV can be approximately regarded as stationary. Denote $l_U\left(m\right) = \left(x_U\left(m\right), y_U\left(m\right)\right)$ as the two-dimensional position of UAV in time slot m, then the flight trajectory of can be approximated by the sequence $\left\{l_U\left(m\right)\right\}_{m=1}^{M}$. Since the UAV flies at the constant speed v, then $\left\|l_U\left(m\right) - l_U\left(m-1\right)\right\| = \delta_t v, m = 2, ..., M$, where the operator $\left\|\alpha\right\|$ means the Euclidean norm of vector α. During the flight, it is necessary to ensure that there will be no collision. In order to simplify the model, the airspace occupied by obstacles is approximately regarded as a circular region with radius R, and denoted by Ω. Generally, the mobile distance of UAV in a time slot is far less than the radius R i.e.$\delta_t v \ll R$. Therefore, when $l_U\left(m\right) \notin \Omega, \forall m = \left[1, 2, ..., M\right]$ is satisfied, the UAV flight path will not pass through the obstacle area, and there will be no collision.

B. Channel Model

Referring to the 3GPP specification [30], the path loss of the communication link between UAV and its serving UE is randomly determined by line-of-sight (LoS) and non-line-of-sight(NLoS) links according to probability. This probability depends on the UAV flight altitude H, the distance between the UAV and connected UE $d_k\left(m\right) = \sqrt{H^2 + \left\|l_U\left(m\right) - l_k\right\|^2}, \forall k \in \mathcal{K}$ and the carrier frequency f_c.

Specifically, the path loss of the LoS and NLOS links between the UAV and the k-UE is calculated by

$$L_k\left(m\right) = \begin{cases} 30.9 + \left(22.25 - 0.5\log_{10}H\right)\log_{10}d_k\left(m\right) + 20\log_{10}f_c, if\ \ LoS\ link, \\ \max\left\{L_k^{LOS}, 32.4 + \left(43.2 - 7.6\log_{10}H\right)\log_{10}d_k\left(m\right) + 20\log_{10}f_c\right\}, if\ \ NLoS\ link. \end{cases} \tag{1}$$

The probability of the LOS link denoted by Pr_{LOS} is given in

$$Pr_{LoS} = \begin{cases} 1, if\sqrt{{d_k}^2 - H^2} \le d_0, \\ \frac{d_0}{\sqrt{{d_k}^2 - H^2}} + \exp\left\{\left(\frac{-\sqrt{{d_k}^2 - H^2}}{p1}\right)\left(1 - \frac{d_0}{\sqrt{{d_k}^2 - H^2}}\right)\right\}, if\sqrt{{d_k}^2 - H^2} > d_0, \end{cases} \tag{2}$$

where $d_0 = \max\left[294.05\log_{10}H - 432.94, 18\right]$ and $p1 = 233.98\log_{10}H - 0.95$. Then the probability of NLOS link is obtained naturally as $\mathrm{Pr}_{NLOS} = 1 - \mathrm{Pr}_{LOS}$.

According to the above path loss model, the channel gain between the UAV and the k-th UE in the time slot m is

$$\mathrm{g}_k\left(m\right) = Pr_k^{LOS}\left(m\right)\left[10^{L_k^{LOS}\left(m\right)/10}\right]^{-1} + \left(1 - Pr_k^{LOS}\left(m\right)\right)\left[10^{L_k^{NLOS}\left(m\right)/10}\right]^{-1}. \tag{3}$$

C. Transmission Model

For simplicity but without loss of generality, we assume that the transmission data size of each UE is F bits in the post disaster rescue scenario. We define the effective radiation angle of the UAV BS antenna as θ, then the maximum distance between the accessible UE and the UAV is $H/\cos\theta$. The above channel model shows that the channel gain $\mathrm{g}_k(m)$ is negatively related to distance $d_k(m)$. It means that if a UE is in the coverage of the UAV BS, the channel gain, the signal-to-noise ratio (SNR) as well, is larger than a certain value. Therefore, the definition of effective radiation angle θ is used as a parameter to make sure that only when UEs' SNR reaches a certain threshold, these UEs can access the UAV to upload data. According to the location of UAV $l_U(m)$, the location of UE $l_k \in \mathbb{R}^{2\times 1}, k \in \mathcal{K}$ and the radiation angle θ, the set of UEs within the coverage of the UAV in time slot m is determined as $\mathcal{K}_{\text{cover}}(m) = \{k \in \mathcal{K} : d_k(m) \le H/\cos\theta\}$. Denote the UE access indicator by $a_k(m)$. $a_k(m) = 1$ indicates that the k-th UE is connected with the UAV in time slot m, conversely $a_k(m) = 0$ means that the k-th UE is not accessed. Thus, the set of UEs associated with the serving UAV in time slot m is expressed as $\mathcal{K}_{com}(m) = \{k \in \mathcal{K} : a_k(m) = 1\}$. Denote $N(m) = \|\mathcal{K}_{com}(m)\|_0$ as the number of UEs in the set $\mathcal{K}_{com}(m)$.

The UAV communication networks employs orthogonal frequency division multiple access (OFDMA) for multiple UEs accessing, so the inter-frequency interference among UEs can be ignored. Then, according to Shannon's Theorem, the transmission rate from UE k to the UAV is

$$R_k(m) = a_k(m)\, B_W \log_2\left(1 + \frac{g_k(m)\, P_{Tx}}{\sigma^2}\right)/N(m), \tag{4}$$

where B_W is the available frequency bandwidth of the system, P_{Tx} is the transmission power of UEs, and σ^2 represents the power of Additive White Gaussian Noise (AWGN) at the UAV receiver.

Therefore, for UE k , the uploaded data size in time slot m can be expressed as

$$w_k(m) = R_k(m)\, \delta_t. \tag{5}$$

Let $W_k(m)$ represents the total bits the k-th UE has uploaded before the m-th time slot, $W_k(m) = \sum\limits_{i=1}^{m} w_k(i)$. Then, the UE access indicator $a_k(m)$ is determined by the distance $d_k(m)$ and $W_k(m)$. If $k \in \mathcal{K}_{\text{cover}}$ and $W_k(m) < F$, $a_k(m) = 1$, otherwise, $a_k(m) = 0$.

D. Energy Model

The energy consumption of UE consists two parts, energy consumption in transmission model and energy consumption in sleep model. We omit the energy consumption in the shift between the transmission and sleep model. So the energy consumption of UE k in time slot m is

$$e_k\left(m\right) = a_k\left(m\right) P_{Tx}\delta_t + \left(1 - a_k\left(m\right)\right) E_{Sleep}, \tag{6}$$

where E_{Sleep} is the energy consumption of UE k in sleep model in time slot m.

E. Problem Formulation

Our goal is to collect the information of users in the area as much as possible, so as to improve the success rate of rescue and reduce casualties. It is worth noting that the energy of UE is very valuable due to the paralysis of ground power system and limited user mobility after the disaster. In addition, there are obstacles that affect the UAV flight. Once the UAV comes into collision with those obstacles, the communication may be interrupted, even out of service. Therefore, we formulate the constrained optimization problem to maximizing the long-term uplink throughput via UAV flight trajectory design. Based on above models, the optimization problem is

$$\text{(P1):} \quad \max_{\{l_U(m)\}_{m=1}^{M}} \frac{1}{T} \sum_{m=1}^{M} \sum_{k=1}^{K} w_k\left(m\right), \tag{7}$$

$$\text{s.t.} \quad \sum_{m=1}^{M} e_k\left(m\right) \le e_0, \ \forall \ k \in K, \tag{7a}$$

$$\left\| l_U\left(m+1\right) - l_U\left(m\right) \right\| = \delta_t v, m = 1, ..., M, \tag{7b}$$

$$l_U\left(m\right) \notin \Omega, m = 1, 2, ..., M. \tag{7c}$$

Constraint (7a) represents that the maximum energy available of each UE is e_0; constraint (7b) means the flight speed of UAV is fixed as v; constraint (7c) guarantees that UAV will not collide with obstacles.

We notice that P1 is a dynamic optimization problem aiming at maximizing the long-term throughput of the system. What's more, the left side of (7a) is also a long-term cumulative variable related to UAV flight trajectory. This means that the whole flight process needs to be taken into account when solving the position of UAV in a certain time slot, which makes it difficult to solve P1 by traditional optimization methods.

III. Safe-DQN Based UAV Trajectory Optimization Algorithm

Since the position of UAV at time slot $m+1$ only depends on the position and moving direction at time slot m, its flight process can be regarded as a discrete-time Markov Decision Process with the UAV as an agent. In this section, we transform the problem (7) coupled with constraints into a Constrained Markov Decision Process (CMDP). For the constraint (7a), we first propose a Lyapunov function based method to determine the set of safe policies. Then, a model-free deep reinforcement learning algorithm, safe-DQN, is adopted to tackle the long-term cost constraint. For the constraint (7c), we define the concept of legal action, which is used to avoid obstacles by judging whether the action is legal before executing it.

A. CMDP Model

CMDP is a typical framework for constrained reinforcement learning tasks. In this framework, the agent needs to maximize a long-term reward while satisfying cost constraints. It is worth noting that, unlike general constraints, the cost constraint in CMDP is long-term and global [31]. As (7a) contains K inequality, the corresponding CMDP will have K cost functions, which makes the solution very complicated. In order to simplify, we transform it into one inequality as follows,

$$\max_{k\in K}\left\{\sum_{m=1}^{M} e_k\left(m\right)\right\} \leq e_0. \tag{7a'}$$

$(7a')$ represents that the maximum value among UEs energy consumption can not exceed e_0. It's obvious that $(7a')$ is a necessary and sufficient condition for (7a), and thus they are equivalent. However, $(7a')$ is no longer a form of time slot summation, which does not meet the requirements for cost function of CMDP. So we exchange the order of summing and taking the maximum value, and get

$$\sum_{m=1}^{M} \max_{k\in K}\left\{e_k\left(m\right)\right\} \leq e_0. \tag{7a''}$$

As $(7a'')$ is a sufficient condition for $(7a')$, (7) is transformed as

$$\begin{aligned} &\max_{\{l_U(m)\}_{m=1}^{M}} \frac{1}{T}\sum_{m=1}^{M}\sum_{k=1}^{K} w_k\left(m\right), \\ &\text{s.t.} \quad (7a''),\ (7b),\ (7c). \end{aligned} \tag{8}$$

and then we can transform (8) to a CMDP. There are seven basic elements in CMDP $\{S, A, w, e, P, s_0, e_0\}$, which are defined as follows in our model:

- S is the state space. In our maximization problem, the state in time slot m consists of the UAV position $l_U(m)$ and the uploaded bits by UE k, $W_k(m)$.
- A is the action space. We define the action as the flight direction of the UAV. As the length of time slot δ_t is small enough, we can discretize the flight direction reasonably without great influence on the final path, and only consider five flight directions including front, back, left, right and hovering.
- w is the instantaneous reward which is defined as the size of data collected in the system in time slot m.
$$w(s_m) = \sum_{k=1}^{K} w_k(s_m), m = 1, 2, ..., M. \tag{9}$$
- e is the instantaneous cost, which is defined as the maximum value of energy consumption among UEs in time slot m.
$$e(s_m) = \max_{k \in \mathcal{K}} \{e_k(s_m)\}, m = 1, 2, ..., M. \tag{10}$$
- P represents the state transition probability matrix. In our optimization problem, the state space is large, and it is very difficult to predict the probability of state transition. For this kind of MDP problem in which the knowledge about P is not priori, model-free reinforcement learning is one of effective solutions.
- The initial state $s_0 \in S$ consists of the starting point of the UAV which is known and fixed and the bits which have been uploaded at the beginning (zeros naturally).
- e_0 is the upper bound of the cumulative cost, which is defined as the energy available to UE in our model.

We define the policy set in the m-th time slot as $\Delta(s_m) = \left\{\pi(\cdot|s_m) \,|\, \sum_{a \in A} \pi(\cdot|s_m) = 1\right\}, \forall s_m \in S$. It can be seen from the definition that the strategy is actually a set of vectors representing the probability of each action being selected in state s_m. For a given strategy $\pi \in \Delta$ and the initial state s_0, the long-term cumulative reward, that is, the total uploaded bits during the flight time T, is expressed as
$$W_\pi(s_0) = \mathbb{E}\left[\sum\nolimits_{m=0}^{M-1} w(s_m) \,|s_0, \pi\right]. \tag{11}$$
Similarly, the long-term cumulative cost, i.e. the left side of $(7a'')$, is
$$E_\pi(s_0) = \mathbb{E}\left[\sum\nolimits_{m=0}^{M-1} e(s_m) \,|s_0, \pi\right]. \tag{12}$$

By constructing CMDP, the position of UAV in time slot $m+1$ is completely determined by the position and flight direction in time slot m, and $l_U(m)$, $l_U(m+1)$ always satisfy the constraint (7b). The flight time T is a constant. Thus, the optimization problem (8) is equivalent to: given s_0 and e_0, find the optimal strategy π^* to maximize the long-term reward while satisfying $E_\pi\left(s_0\right) \leq e_0$ and (7c), that is, solve the problem as follows,

$$\begin{aligned} &\text{(P2):}\ \max_{\pi\in\Delta}\left\{W_\pi\left(s_0\right): E_\pi\left(s_0\right)\leq e_0\right\}, \\ &\text{s.t.}\quad (7c). \end{aligned} \tag{13}$$

B. Lyapunov Function Based Safe Policy Set

In this subsection, we leave (7c) out of the question temporarily, which is tackled in next subsection. Then, the key to solving (13) is to determine the set of ”safe” strategies that meet the condition $E_\pi\left(s_0\right) \leq e_0$ and select the optimal policy from it. For this, we adopt following Lyapunov function based method to determine the set of safe policies [32].

For the convenience of representation, we introduce a general Bellman operator, which consists of a policy π and a general reward function (or cost function) h,

$$T_{\pi,h}[V](s)=\sum\nolimits_{a}\pi\left(a|s\right)\Big[h\left(s\right)+\sum\nolimits_{s'\in S'}P(s|s',a)V(s')\Big], \tag{14}$$

where s' is the next state of $s \in S$ under the action $a \in A$. It can be seen that $T_{\pi,h}\left[V\right]\left(s\right)$ is a function that describes the long-term cumulative expected value. When h is the reward function w, $W_\pi\left(s_0\right) = T_{\pi,w}\left[W\right]\left(s_0\right)$; when h is the cost function e, $E_\pi\left(s_0\right) = T_{\pi,e}\left[E\right]\left(s_0\right)$.

We assume a benchmark policy $\pi_B \in \Delta$ and define a set of Lyapunov candidate functions

$$L_{\pi_B}\left(s_0, e_0\right) = \{L : T_{\pi_B,e}\left[L\right]\left(s\right) \leq L\left(s\right), \forall s \in S; L\left(s_{M-1}\right) = 0; L\left(s_0\right) \leq e_0\}, \tag{15}$$

where s_{M-1} is the last state, that is, the landing position of the UAV, which is fixed and known in our model. Consider the cumulative cost function $E_{\pi_B}\left(s\right)$ with the benchmark policy. It satisfies all requirements for Lyapunov function in (15), that is, $E_{\pi_B}\left(s_0\right) \leq e_0$, $E_{\pi_B}\left(s_{M-1}\right) = 0$, and $E_{\pi_B}\left(s\right) = T_{\pi_B,e}\left[E_{\pi_B}\right]\left(s\right) = \mathbb{E}\left[\sum_{m=0}^{M-1} e\left(s_m\right)|s_0, \pi_B\right]$. Therefore, the set of Lyapunov candidate functions defined in (15) must be non-empty. Corresponding to any Lyapunov function $L\left(s\right) \in L_{\pi_B}\left(s_0, e_0\right)$, there exists a set of safe strategies

$$F_L\left(s\right) = \left\{\pi\left(\cdot|s\right) \in \Delta : T_{\pi,e}\left[L\right]\left(s\right) \leq L\left(s\right)\right\}. \tag{16}$$

In order to ensure that the safe strategies set contains the optimal solution of the problem π^*, the constructed Lyapunov function should not only satisfy the three conditions in (15), but also satisfy

$$T_{\pi^*,e}\left[L\right]\left(s\right) \leq L\left(s\right). \tag{17}$$

According to the **Lemma 1.** in [32], there is an auxiliary cost function $\varepsilon\left(s\right)$ such that the Lyapunov function conforming to (15) and (17) can be expressed as

$$L_{\varepsilon}\left(s\right)=\mathbb{E}\left[\sum\nolimits_{m=0}^{M-1}e\left(s_{m}\right)+\varepsilon\left(s_{m}\right)|\pi_{B},s\right], \tag{18}$$

and $L_{\varepsilon}\left(s\right)$ is equal to the cumulative cost function under the optimal strategy, that is $L_{\varepsilon}\left(s\right)\in L_{\pi_{B}}\left(s_{0},e_{0}\right)$ and $L_{\varepsilon}\left(s\right)=E_{\pi^{*}}\left(s\right)$. However, as the optimal policy π^{*} is not priori, it is difficult to construct a suitable $\varepsilon\left(s\right)$ directly. Therefore, we adopt the method proposed in [32] to approximate the auxiliary cost $\varepsilon\left(s\right)$ to a constant function, which is independent of state,

$$\tilde{\varepsilon}=\frac{\left(e_{0}-E_{\pi_{B}}\left(s_{0}\right)\right)}{\mathbb{E}\left[\mathrm{T}^{*}|s_{0},\pi_{B}\right]},\forall s_{0}\in S, \tag{19}$$

where $\mathbb{E}\left[\mathrm{T}^{*}|s_{0},\pi_{B}\right]$ is the expected stopping time of the CMDP. In our problem, the working time of UAV is certain, that is $\mathbb{E}\left[\mathrm{T}^{*}|s_{0},\pi_{B}\right]{=}M$. Hence, (19) is

$$\tilde{\varepsilon}=\frac{1}{M}\left(e_{0}-E_{\pi_{B}}\left(s_{0}\right)\right). \tag{20}$$

Substituting (20) into (18) , we can get the Lyapunov function as

$$L_{\tilde{\varepsilon}}\left(s\right)=\mathbb{E}\left[\sum\nolimits_{m=0}^{M-1}e\left(s_{m}\right)+\tilde{\varepsilon}|\pi_{B},s\right]. \tag{21}$$

and the corresponding safe policy set defined in (16) is

$$F_{L_{\tilde{\varepsilon}}}\left(s\right)=\left\{\pi\left(\cdot|s\right)\in\Delta:T_{\pi,e}\left[L_{\tilde{\varepsilon}}\right]\left(s\right)\leq L_{\tilde{\varepsilon}}\left(s\right)\right\}. \tag{22}$$

Therefore, with the help of Lyapunov function, P2 of (13) without constraint (7c) is equivalently described as

$$\pi^{*}\left(\cdot|s\right)=\arg\max_{\pi\in F_{L_{\tilde{\varepsilon}}}\left(s\right)}W_{\pi}\left(s_{0}\right),\forall s\in S. \tag{23}$$

To sum up, in this subsection, we construct the appropriate Lyapunov function $L_{\tilde{\varepsilon}}\left(s\right)$ by introducing the auxiliary cost function $\tilde{\varepsilon}$. Then, based on $L_{\tilde{\varepsilon}}\left(s\right)$, we determine the set of safe policies satisfying the constraint ($7a''$), which lays foundation for the following subsection to solve the optimal policy.

C. Deep Reinforcement Learning Based Solution For CMDP: safe-DQN

In CMDP $\left\{S,A,w,e,P,s_{0},e_{0}\right\}$, the next state is determined by the current state and action. Therefore, when the agent chooses an action, it needs to consider not only the immediate returns and costs, but also the impact on the future. Based on above considerations, the state-action reward function ($S \times A \to R$) is defined as

$$Q_w\left(s_m, \mathrm{a}_m\right) = \mathbb{E}\left[\sum_{t=m}^{M-1-m} \gamma^{t-m} w\left(s_t\right)|s_0, a_0\right], \forall s_m \in S, a_m \in A, \tag{24}$$

where $\gamma \in [0,1]$ is the discount factor, which represents that the influence of future rewards on the current value function decays exponentially. Using the Behrman operator, (24) is rewritten as

$$Q_w\left(s, a\right) = w\left(s\right) + \gamma V_w^\pi\left(s'\right), \forall s \in S, a \in A, \tag{25}$$

where $V_w^\pi\left(s\right) = w\left(s\right) + \gamma \sum\limits_{s' \in S} P_{s_m, \pi\left(s_m\right)}\left(s'\right) V_w^\pi\left(s'\right), \forall s \in S$. Similarly, the state-action cost function is

$$Q_e\left(s, a\right) = e\left(s\right) + \gamma V_e^\pi\left(s'\right), \forall s \in S, a \in A, \tag{26}$$

where $V_e^\pi\left(s\right) = e\left(s\right) + \gamma \sum\limits_{s' \in S} P_{s', \pi\left(s'\right)}\left(s'\right) V_e^\pi\left(s'\right), \forall s \in S$. And the Lyapunov function (21) is expressed as

$$Q_l\left(s, a\right) = e\left(s\right) + \tilde{\varepsilon} + \gamma V_l^\pi\left(s'\right), \forall s \in S, a \in A, \tag{27}$$

where $V_l^\pi\left(s\right) = e\left(s\right) + \tilde{\varepsilon} + \gamma \sum\limits_{s' \in S} P_{s', \pi\left(s'\right)}\left(s'\right) V_l^\pi\left(s'\right), \forall s \in S$.

Observing and analyzing (25)-(27), we can rewrite (27) as

$$Q_l\left(s, a\right) = Q_e\left(s, a\right) + \tilde{\varepsilon} Q_T\left(s\right), \forall s \in S, a \in A, \tag{28}$$

where $Q_T\left(s_m\right) = \sum\limits_{t=m}^{M-1-m} \gamma^{t-m}, \forall s_m \in S$ is a function related to the number of remaining steps and the discount factor, and can be directly obtained by calculation.

If $Q_w\left(s, a\right)$ and $Q_e\left(s, a\right)$ are known, according to (19), the auxiliary cost under the benchmark strategy π_B can be calculated by

$$\varepsilon' = \frac{e_0 - \pi_B\left(\cdot|s_0\right)^\top Q_e\left(s_0, \cdot\right)}{\pi_B\left(\cdot|s_0\right)^\top Q_T\left(s_0\right)}, \tag{29}$$

and the set of safe policies (22) is

$$F_{Q_l}\left(s\right) = \{\pi\left(\cdot|s\right) \in \Delta : \left(\pi\left(\cdot|s\right) - \pi_B\left(\cdot|s\right)\right)^\top Q_l\left(s, \cdot\right) \le \tilde{\varepsilon}\ \}. \tag{30}$$

Then (23) can be expressed as finding the optimal strategy

$$\pi^*\left(\cdot|s\right) = \arg \max_{\pi\left(\cdot|s\right) \in F_{Q_l}\left(s\right)} \pi\left(\cdot|s\right)^\top Q_w\left(s, \cdot\right), \forall s \in S, \tag{31}$$

that is, solving the following linear programming problem.

$$\pi^*\left(\cdot|s\right) \in \arg \underset{\pi \in \Delta}{max} \{\pi\left(\cdot|s\right)^\top Q_w\left(s, \cdot\right) : \left(\pi\left(\cdot|s\right) - \pi_B\left(\cdot|s\right)\right)^\top Q_l\left(s, \cdot\right) \le \varepsilon'\}. \tag{32}$$

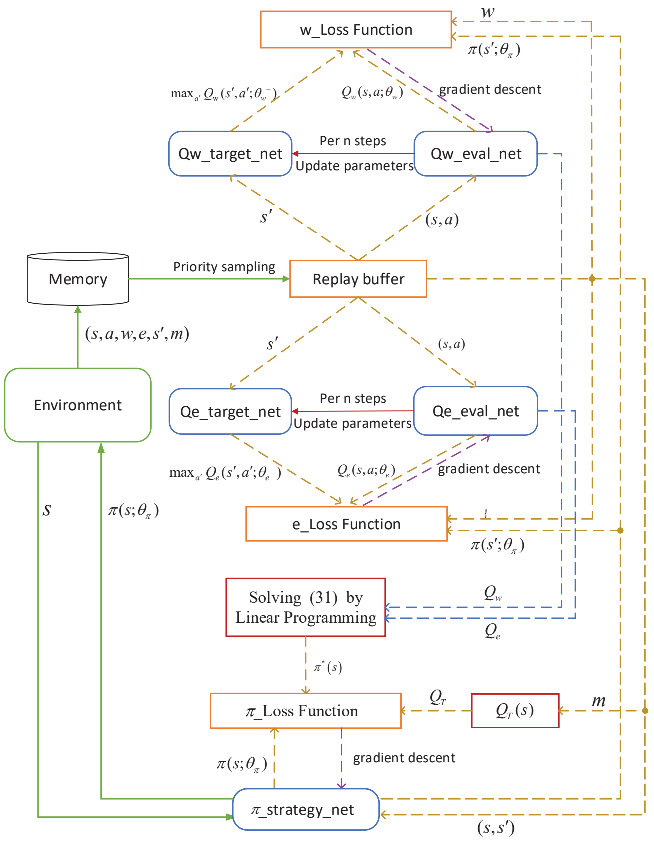


Fig. 2: The block diagram of the safe-DQN algorithm

Solving (32) requires accurate calculation of $Q_w(s,a)$, $Q_e(s,a)$ and $\pi_B(\cdot|s)$. However, due to the complex nonlinear relationship between state, action and the value functions, it is almost impossible to obtain the mathematical expression of them directly. Reinforcement learning is one of the effective ways to establish mapping relationship. In common reinforcement learning algorithms, sarsa and Q-Learning obtain the optimal strategy by constructing and maintaining a state-action value table, where each state-action tuple corresponds to a value, so they can only solve problems which have a small number of states and actions. Deep Q-network is an improvement of Q-learning. It estimates the value function through a deep neural network, which can solve the situation of a large number of states but cannot cope with a large action space. The policy-based policy gradient algorithm can solve continuous state and action by constructing a policy network to directly output actions, but the network can only be updated in rounds, which makes a low training efficiency. Actor-critic and deep deterministic policy gradient algorithms combine policy-based and value-based methods, which can not only deal with an infinite number of states and actions, but also ensure network convergence. At the same time, they have higher computational complexity compared with other reinforcement learning algorithms. In the CMDP $\{S, A, w, e, P, s_0, e_0\}$ problem we constructed, the action space is small (five dimensions) but the number of states is large. Thus, considering the applicability and complexity of these algorithms comprehensively, we adopt a model-free safe-DQN algorithm to solve (32). The block diagram of safe-DQN is shown in Fig.2.

First of all, we build two sets of DQN networks and output $\hat{Q}_w\left(s, a, \theta_w\right)$, $\hat{Q}_e\left(s, a, \theta_e\right)$ to approximate $Q_w\left(s, a\right)$ and $Q_e\left(s, a\right)$ respectively. That is $Q_w\left(s, a\right) \approx \hat{Q}_w\left(s, a, \theta_w\right)$, $Q_e\left(s, a\right) \approx \hat{Q}_e\left(s, a, \theta_e\right)$, where θ_w and θ_e are the parameters of the reward network and the cost network respectively. In the DQN algorithm we adopt, in order to remove the correlation between samples, the experience playback mechanism is introduced; in order to reduce the correlation between the real Q value and the output of neural networks, two neural networks with the completely same structure are used, one for estimated value, and the other for target value.

Taking the approximate network of the state-action reward function as an example, the estimated value network $\hat{Q}_w\left(s, a; \theta_w\right)$ needs to update its parameters continuously through training, while the target value network $\hat{Q}_w\left(s, a; {\theta_w}^-\right)$ is only used to calculate the value of the reward function at next state and its parameters don't need to be updated iteratively, but are copied from the estimated value network at intervals. In each iteration, a certain number of samples $B = \{(s_j, a_j, w_j, e_j, {s_j}', g_{w,j}, g_{e,j})\}_{j=1}^{|B|}$ are selected from the memory according to their priority $\{(g_{w,j})\}\,|_{j=1}^{|B|}$, which are determined by their TD-errors

$$\left\{\mathrm{y}_j^w - \hat{Q}_w\left(s_j, a_j; \theta_w\right)\right\}_{j=1}^{|B|}. \tag{33}$$

$\mathrm{y}_j^w = w_j + \gamma\pi(\cdot|{s_j}')^\top \hat{Q}_\mathrm{w}\left({s_j}', ; {\theta_j}^-\right)$ represents the target reward value of the j sample, which is calculated by the immediate reward, the output of the target value network at next state and the policy of the next state. Then, the loss function of the reward network is calculated by

$$Loss\left(\theta_w\right) = \frac{1}{B}\sum_{\mathrm{j}=1}^{B} w_{w,j}\left(\mathrm{y}_j^w - \hat{Q}_w\left(s_j, a_j; \theta_w\right)\right)^2. \tag{34}$$

Finally, the parameters θ_w are updated by gradient back propagation of the neural network with specific learning-rate α, as

$$\theta_w = {\theta_w}^- - \alpha\nabla_{\theta_w} Loss\left(\theta_w\right). \tag{35}$$

Remark 1. *The learning-rate α is the stepsize when the network parameters are updated with gradient descent, which determines the distance of parameters alteration in each iteration. Larger*

α is likely to cause the algorithm to oscillate greatly near local optimum and is difficult to converge. Smaller α makes the parameters change little in each iteration, which leads to a slow convergence speed of the algorithm.In order to balance the stability and convergence speed of the algorithm, we often need to try time and again to find a compromise α.

Similarly, in the approximate network of the state-action cost function, the TD-errors of the samples are

$$\left\{ \mathrm{y}_j^e - \hat{Q}_e\left(s_j, a_j; \theta_e\right) \right\}_{j=1}^{|B|}, \tag{36}$$

where $\mathrm{y}_j^e = e_j + \gamma\pi(\cdot|s_j{}')^\top \hat{Q}_e\left(s_j{}', ; \theta_e{}^-\right)$ represents the target cost value of the j sample. The loss function of the cost network is calculated by

$$Loss\left(\theta_e\right) = \frac{1}{B}\sum_{j=1}^{B} g_{e,j}\Big(\mathrm{y}_j^e - \hat{Q}_e\left(s_j, a_j; \theta_e\right)\Big)^2, \tag{37}$$

and the parameters θ_e are updated according to

$$\theta_e = \theta_e{}^- - \alpha\nabla_{\theta_e} Loss\left(\theta_e\right). \tag{38}$$

In addition to approximating $Q_w\left(s, a\right)$ and $Q_e\left(s, a\right)$, a reasonable value for the benchmark strategy $\pi_B\left(\cdot|s\right)$ is needed to solve the problem (32). However, due to the unpredictability of the future and the large dimension of the state space, it is very difficult to directly determine a benchmark strategy that meets the conditions. To this end, we build a deep neural network (DNN) to parameterize the policy and approximate the value of the benchmark strategy with the output of the DNN, namely $\pi_B\left(\cdot|s\right) \approx \hat{\pi}\left(\cdot|s;\theta_\pi\right)$. In each iteration, the parameters θ_π are updated by reducing the loss function of the policy network. As given in

$$L\left(\theta_\pi\right) = \mathbb{E}_{(s_j)\sim B}\left[D_{KL}\left(\hat{\pi}\left(\cdot|s_j;\theta_\pi\right)||\pi^*\left(\cdot|s_j\right)\right)\right], \tag{39}$$

the loss function is defined as the KL divergence between the benchmark strategy and the optimal strategy, which represents the difference between the two policy vector distributions. The optimal strategy $\pi^*\left(\cdot|s_j\right)$ is obtained by solving the linear programming problem (32) with the approximate benchmark strategy $\hat{\pi}\left(\cdot|s_j;\theta_\pi\right)$. The parameters θ_π are updated according to

$$\theta_\pi \leftarrow \theta_\pi - \alpha\nabla_{\theta_\pi} L\left(\theta_\pi\right). \tag{40}$$

With the reward function network, the cost function network and the policy network, the ε' in (29) is approximated to

$$\hat{\varepsilon}' = \frac{e_0 - \hat{\pi}(\cdot|s_0;\theta_\pi)^\top \hat{Q}_e\left(s_0, \cdot; \theta_e\right)}{\hat{\pi}(\cdot|s_0;\theta_\pi)^\top Q_T\left(s_0\right)}. \tag{41}$$

In summary, in each iteration of safe-DQN, three networks are trained in sequence, and finally the optimal policy that meets the "safe" condition $(7a'')$ can be obtained.

All of the above are proposed to tackle the constraint $(7a'')$. For the constraint (7c), we adopt a simple judgment method. We propose the concept of legal actions, which ensure that the UAV is outside the obstacle area in the next time slot. The set of legal actions in each state is $A_{\text{legal}}\left(\mathrm{m}\right)=\left\{a \in A : l_U\left(m+1\right) \notin \Omega\right\}$. In each time slot, before the action is executed, the UAV needs to judge whether the action is legal, and if it is not, another legal action will be selected randomly. Besides, in order to ensure the effectiveness of learning, samples with illegal actions will not be stored in the memory of safe-DQN.

The detailed procedure of the proposed safe-DQN based trajectory design algorithm is given as follows.

D. Analysis of the Proposed Algorithm

1. Complexity: Denote $|S|$ as the size of state space, and $|A|$ as the size of action space. Assume the algorithm converges within D iterations. In each iteration, three networks $\hat{Q}_w$,$\hat{Q}_e$,π_θ need to be updated and the computational complexity of each network is $O\left(|S||A|\right)$. Secondly, there are $|S|$ linear programming problems to be solved and each of them has $|A|$ decision variables and $(|A|+1)$ constraint conditions, so its complexity is $O(|S||A|^2(|A|+1))$. Thus, the computational complexity of proposed algorithm is $O\left(3D|S||A|+D|S||A|^2(|A|+1)\right) \approx O\left(3D|S||A|+D|S||A|^3\right)$. Generally speaking, the number of iterations needed for convergence is far less than $|S||A|$. Therefore, the complexity of the safe-DQN based algorithm is much less than that of polynomial time algorithm $O\left(|S|^2|A|^2(|S||A|(|A|+1)\right)$ [33].

2. The "safe" property: Different from traditional reinforcement learning algorithms, safe-DQN is able to solve the optimization problem with dynamic and long-term accumulation constraints with the help of Lyapunov function. Compared with the general deep Q-network algorithm, the safe-DQN has higher complexity, but its safe property has great significance in solving practical problems.

IV. Simulation Results

In this section, the performance verification of the proposed safe-DQN based UAV trajectory design algorithm is presented. It is assumed that a UAV is responsible for searching a pre-allocated area where K affected users are randomly distributed. When the affected area is large,

Algorithm 1 Safe-DQN based trajectory design algorithm

```
Initialization System Parameters: user locations $l_k, k \in \mathcal{K}$; length of time slot $\delta_t$; number
    of time slots $M$; UAV filght speed $v$; UAV flight height $H$; upper limit of UE energy
    consumption $e_0$; obstacle area $\Omega$.
Initialization Algorithm Parameters: prioritized replay buffer $U = \{\emptyset\}$; importance weights
    $g_{w,0} = 1, g_{e,0} = 1$; mini-batch size $|B|$; network parameters $\theta_w{}^-, \theta_e{}^-, \theta_\pi$.
 1: for $k \in \{0, 1, ..., \}$ do
 2:   Initialize UAV position as the take-off point $l_U\left(0\right)$; uploaded bits $w\left(s_0\right) = 0$; UE energy
      consumption $e\left(s_0\right) = 0$.
 3:   for $t = 0$ to $t = M - 1$ do
 4:     Obtain action $a_t$ according to the policy network (DNN) $\hat{\pi}\left(\cdot|s_t;\theta_\pi\right)$.
 5:     if $a_t \in A_{\text{legal}}(t)$ then
 6:       Add this experience to replay buffer,
          $U \leftarrow (s_t, a_t, w_t, e_t, s_{t+1}, g_{w,t}, g_{e,t}) \cup U$,
 7:       From the buffer $U$, sample a mini-batch
          $B = \{(s_j, a_j, w_j, e_j, s_{j+1}, g_{w,j}, g_{e,j})\}_{j=1}^{|B|}$,
 8:       Update the deep Q network (DQN) of state-action reward function $\hat{Q}_w\left(s, a, \theta_w\right)$
          according to (35),
 9:       Update the deep Q network (DQN) of state-action cost function $\hat{Q}_e\left(s, a, \theta_e\right)$ according
          to (38),
10:       Update important weights $g_{w,j}, g_{e,j}$ based on TD-errors given in (33) and (36),
11:       Calculate $Q_l$ according to (27),
12:       Obtain $\{\pi^*\left(\cdot|s_{\text{j}}\right)\}_{j=1}^{|B|}$ by solving (32),
13:       Update the network of policy $\hat{\pi}\left(\cdot|s;\theta_\pi\right)$ according to (40).
14:     else
15:       Select action $a_t$ from $A_{\text{legal}}(\mathrm{t})$ randomly and then back to step (6).
16:     end if
17:   end for
18:   Update $\theta_w{}^- = \theta_w, \theta_e{}^- = \theta_e$ after $t$ iterations.
19: end for
```

we can deploy multiple UAVs and each of them is responsible for the search and rescue work in the pre-determined small area. The detailed simulation parameters are shown in Table I.

TABLE I: Simulation Parameters

Parameter	**Value**
Flight altitude	H = 100 m
Flight speed	v = 30 m/s
Carrier frequency	f_c = 2 GHz
Radio bandwidth	20 MHz
Radius of obstacle area	R = 30 m
Effective angle of UAV radiation	θ = π /8 rad
Time slot length	δ_t = 0.5 s
Transmitting power of UE	P_{Tx} = 23 dBm
Noise power spectral density	-174 dBm/Hz
Standby energy consumption of UE per time slot	E_{base} = 0.01 J

We verify the convergence of the proposed algorithm with different learning-rates in Fig. 3. There are three curves and all of them are simulated under the same condition when $K = 20$ and $T = 100\ s$. As it can be observed, when the learning-rate is set as 0.00005 or 0.000001, the system throughput, i.e, the reward in the CMDP model, gradually increases with the increase of iterations, which indicates that the parameters of neural networks are gradually updated in a good direction. Specifically, when the learning-rate is set as 0.00005, the throughput converges to about 50 Mbps within 1000 episodes. When it is increased to 0.0001, the throughput quickly reaches the maximum value, but performs extremely unstably in the later stage. When the learning-rate is decreased to 0.000001, the growth rate of throughput slows down significantly, and converges to 50 Mbps within 2200 episodes, which is same as the value when learning-rate is 0.00005. This verifies the insights in **Remark 1**, that is, the larger learning-rate makes the network difficult to converge, while the smaller learning-rate makes the network converge stably but the speed is very slow. In order to balance efficiency and stability, we set the learning-rate as 0.00005 in the subsequent simulations.

In order to illustrate the effectiveness of the proposed UAV trajectory design algorithm, we design the following two benchmark algorithms.

1. Shortest flight distance algorithm (SFD): Taking off at the fixed starting point, the UAV

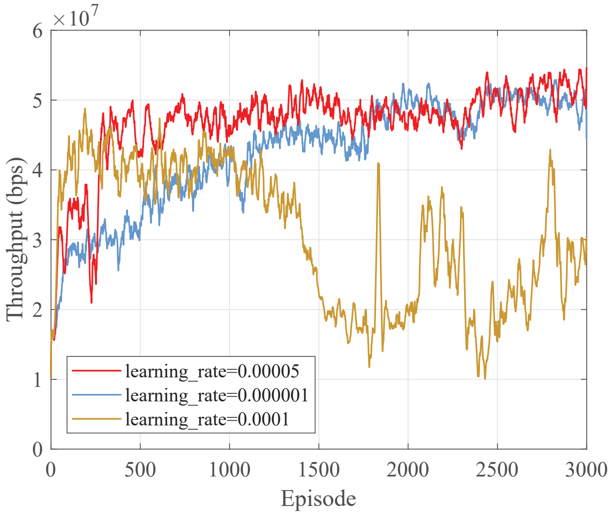


Fig. 3: The convergence of the proposed algorithm with varying learning rates.

selects the one closest to the current location of the UAV among all UEs to be served and then hovers above it to provide communication services. After the transmission is completed, the next location is selected according to the same criteria until the total time T is reached.

2. Fixed flight trajectory algorithm (FFT): The UAV flies along a preestablished path in the affected area, regardless of users' locations.

Next, we investigate the performance of the proposed algorithm compared with benchmark algorithms from Fig. 4 to Fig. 6 with varying service durations and user numbers.

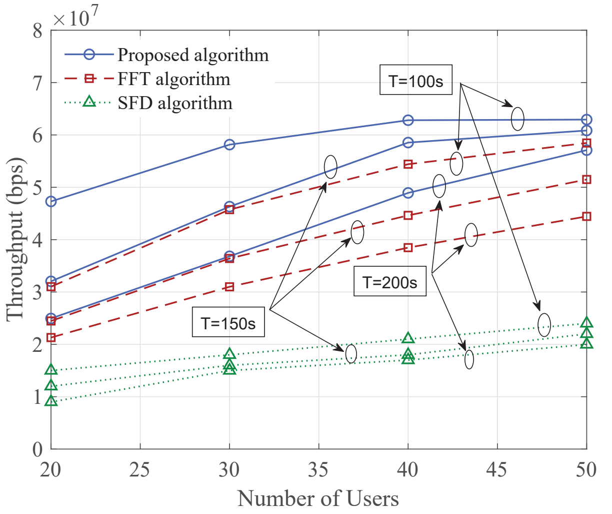


Fig. 4: System throughput with varying user number.

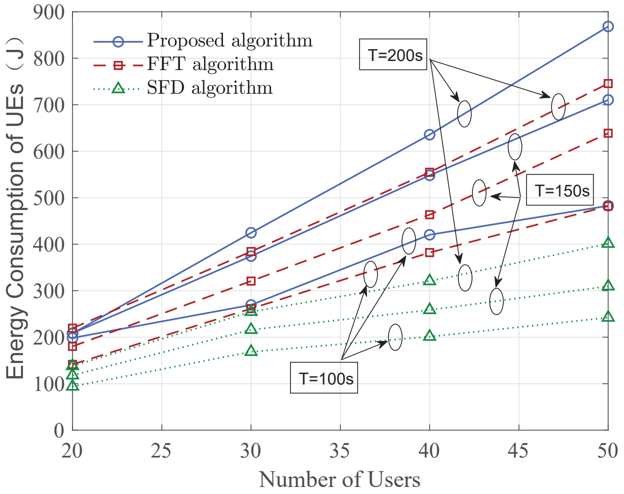
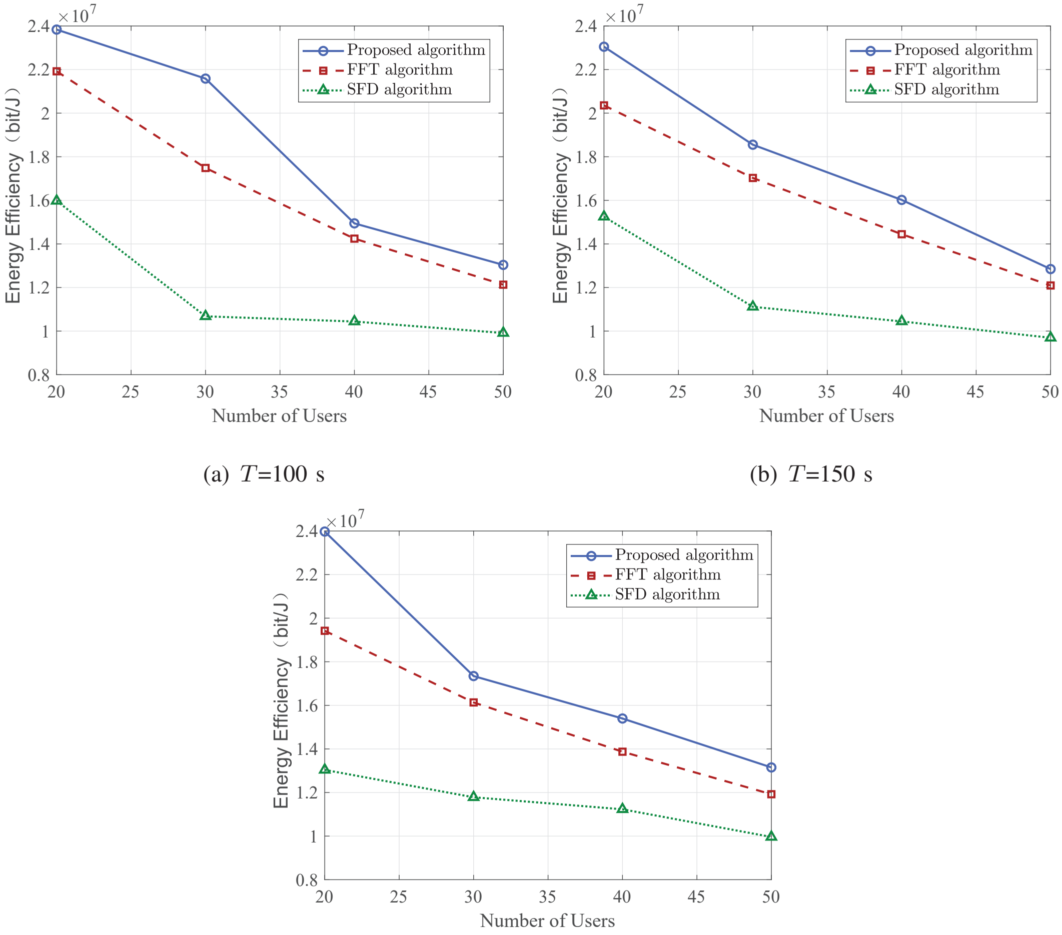


Fig. 5: Energy consumed by UEs with varying user number.

Fig. 4 shows the long-term uplink throughput of the UAV emergency communication network. First of all, each curve in Fig. 4. shows an upward trend, which means that regardless of algorithm and service duration, the system throughput increases with the number of users increasing. However, its growth rate is gradually decreasing. This is due to the fact that the maximum capacity of the communication system with a limited bandwidth is certain. With the increase of K, the system throughput keeps approaching the maximum capacity, but cannot exceed it. Comparing the performance of the same algorithm with different T, it is found that the longer T, the lower the system throughput. This shows that in order to collect user information as much as possible, the UAV emergency communication system needs to sacrifice the time efficiency to a certain extent. Comparing the performance of different algorithms with the same T, we can see that for any K, the proposed algorithm is obviously better than the FFT algorithm, and much better than the SFD algorithm. When $(T = 100\ s, K = 30)$, $(T = 150\ s, K = 40)$, and $(T = 200\ s, K = 50)$, the advantage of the proposed algorithm is more prominent, which is 0.27, 0.31, 0.28 times higher than FFT algorithm and 2.23, 2.25 and 1.89 times higher than SFD algorithm respectively. In addition, for the same T, as the number of users increases, the performance differences among three algorithms change from small to large and then become smaller. This trend is explainable. When there are few users in the area, the demand of the UAV service time is relatively lower, making it not that necessary to optimize the flight trajectory. When there are too many users, the space where the flight path can be optimized is greatly limited because of the tight UAV service time, such as $(T = 100\ s, K = 50)$.



(a) T=100 s

(b) T=150 s

(c) T=200 s

Fig. 6: Energy efficiency of UEs with varying user number.

In Fig. 5, we compare the total energy consumption of UEs among three algorithms. It can be easily inferred that as the number of users increases, the total energy consumption of UEs also continues to increase. For the same algorithm and same K, the longer the UAV works, the more energy is consumed. Comparing the energy consumption of three algorithms with the same T, we find that the energy consumption of the proposed algorithm is always greater than that of FFT algorithm, and even greater than that of the SFD algorithm, which illustrates that the system throughput is increased at the cost of more energy consumption to some extent. Still taking $(T = 100\ s, K = 30)$, $(T = 150\ s, K = 40)$ and $(T = 200\ s, K = 50)$ as examples, the total energy consumption of UEs of the proposed algorithm is increased by 0.03, 0.10, 0.16

times compared with the FFT algorithm and 0.59, 1.09, 1.16 times compared with the SFD algorithm. Obviously, this set of data is less than the increase rate of corresponding throughput.

Based on the analysis of Fig. 4 and Fig. 5, it can be concluded that the proposed algorithm has achieved a large increase in throughput with a little increase in energy consumption. In order to further demonstrate the performance advantages of the proposed algorithm, we compare the energy efficiency (EE) of the three algorithms in Fig. 6. EE is defined as the ratio of the long-term uploaded bits to the total energy consumption of UEs,

$$EE = \frac{\sum\limits_{m=1}^{M}\sum\limits_{k=1}^{K} w_k\left(m\right)}{\sum\limits_{m=1}^{M}\sum\limits_{k=1}^{K} e_k\left(m\right)}. \tag{42}$$

As it can be observed, no matter what values of T and K are set as, the proposed algorithm is able to obtain the maximum energy efficiency, thus effectively improving the network performance.

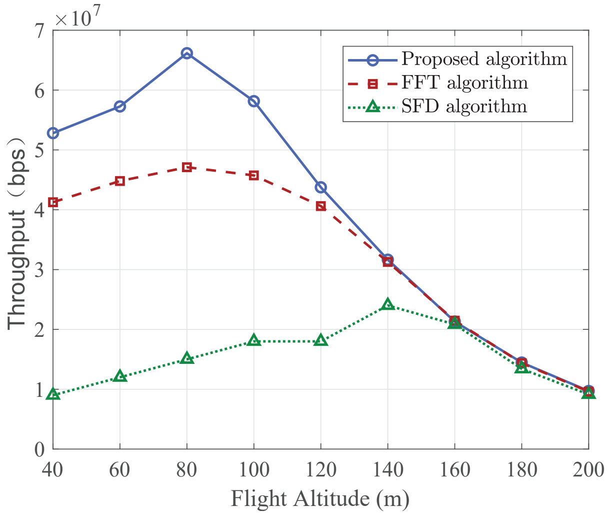


Fig. 7: System throughput with varying flight altitude.

Finally, we discuss the impact of UAV flight height on the system throughput in Fig. 7, where the UAV flight altitude varies from 40 m to 200 m and other parameters remain unchanged. For the proposed safe-DQN based algorithm and the FFT algorithm, the system throughput increases with the UAV altitude changing from 40 m to 80 m. However, the throughput drops rapidly when the height is greater than 80 m. For the SFD algorithm, this inflection point appears when the height is 140 m. The three algorithms all show a trend of increasing first and then decreasing, which reasonably reflects the attenuation characteristics of UAV communication channel. According to (2), the probability of LoS link between the UAV and its serving devices increases with the UAV height increasing. Besides, the path loss of LoS link is less than that of NLoS link. Therefore, the performance of the UAV-based communication network can be improved by increasing the flying height within a certain range. However, when the altitude continues to increase, according to (1), although the rate of increase in $\log_{10} H$ is very slow, the distance between the UAV and the UE is increasing significantly, which ultimately leads to a rapid increase in the path loss and then reduces the system throughput. Therefore, the UAV height needs to be determined reasonably and carefully.

V. Conclusion

In this paper, we studied the trajectory optimization problem in the UAV-based emergency communication networks. The UAV was deployed as mobile aerial base station to collect information from users in affected area. In addition to the limitation of UAV battery, the constraints on UE energy and location of obstacles were also considered. Since the constraint on energy consumption of UE is dynamic and long-term cumulative, we proposed a Lyapunov-based deep learning trajectory design algorithm. The simulation results showed that the proposed algorithm performs better in terms of the system throughput and energy efficiency compared with benchmark algorithms. The algorithm proposed in this paper solved the UAV flight trajectory optimization problem in the case of limited UE energy and flight obstacles. By designing the flight trajectory, the algorithm is able to maximize the system uplink throughput and complete the task of information collection in the post disaster areas. In the case of more ground users or a larger disaster area, multiple UAVs need to be deployed to achieve greater coverage and more user access, which may be included in our future work.

References

[1] R. Munich, "Natcatservice loss events worldwide 1980–2014," *Munich Reinsurance, Munich, Germany*, 2015.

[2] D. G.C., A. Ladas, Y. A. Sambo, H. Pervaiz, C. Politis, and M. A. Imran, "An overview of post-disaster emergency communication systems in the future networks," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 132–139, 2019.

[3] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging UAVs for disaster management," *IEEE Pervasive Comput.*, vol. 16, no. 1, pp. 24–32, 2017.

[4] M. Erdelj and E. Natalizio, "UAV-assisted disaster management: Applications and open issues," in *2016 International Conference on Computing, Networking and Communications (ICNC)*, 2016, pp. 1–5.

[5] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, "Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges," *IEEE Communications Surveys Tutorials*, vol. 21, no. 4, pp. 3417–3442, 2019.

[6] A. Kwasinski, W. W. Weaver, P. L. Chapman, and P. T. Krein, "Telecommunications power plant damage assessment for hurricane katrina - site survey and follow-up results," *IEEE Syst. J.*, vol. 3, no. 3, pp. 277–287, 2009.

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

[8] N. Zhao, X. Pang, Z. Li, Y. Chen, F. Li, Z. Ding, and M. Alouini, "Joint trajectory and precoding optimization for UAV-assisted NOMA networks," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3723–3735, 2019.

[9] T. Zhang, Z. Wang, Y. Liu, W. Xu, and A. Nallanathan, "Caching placement and resource allocation for cache-enabling UAV NOMA networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12 897–12 911, 2020.

[10] Y. Liu, Z. Qin, Y. Cai, Y. Gao, G. Y. Li, and A. Nallanathan, "UAV communications based on non-orthogonal multiple access," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 52–57, 2019.

[11] H. Huang, Y. Yang, H. Wang, Z. Ding, H. Sari, and F. Adachi, "Deep reinforcement learning for UAV navigation through massive MIMO technique," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1117–1121, 2020.

[12] M. Gapeyenko, V. Petrov, D. Moltchanov, S. Andreev, N. Himayat, and Y. Koucheryavy, "Flexible and reliable UAV-assisted backhaul operation in 5G mmWave cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2486–2496, 2018.

[13] X. Liu, Y. Liu, and Y. Chen, "Machine learning empowered trajectory and passive beamforming design in UAV-RIS wireless networks," *IEEE J. Sel. Areas Commun.*, pp. 1–1, 2020.

[14] X. Xu, Y. Zeng, Y. L. Guan, and R. Zhang, "Overcoming endurance issue: UAV-enabled communications with proactive caching," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1231–1244, 2018.

[15] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, 2017.

[16] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for UAV-enabled mobile relaying systems," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4983–4996, 2016.

[17] T. Zhang, Y. Wang, Y. Liu, W. Xu, and A. Nallanathan, "Cache-enabling UAV communications: Network deployment and resource allocation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7470–7483, 2020.

[18] T. Zhang, Y. Xu, J. Loo, D. Yang, and L. Xiao, "Joint computation and communication design for UAV-assisted mobile edge computing in IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5505–5516, 2020.

[19] 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.

[20] Z. Wang, W. Xu, D. Yang, and J. Lin, "Joint trajectory optimization and user scheduling for rotary-wing UAV-enabled wireless powered communication networks," *IEEE Access*, vol. 7, pp. 181 369–181 380, 2019.

[21] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network toward 6G: Machine-learning approaches," *Proc. IEEE*, vol. 108, no. 2, pp. 292–307, 2020.

[22] S. Yin, S. Zhao, Y. Zhao, and F. R. Yu, "Intelligent trajectory design in UAV-aided communications with reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8227–8231, 2019.

[23] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 729–743, 2020.

[24] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, 2018.

[25] X. Liu, Y. Liu, Y. Chen, and L. Hanzo, "Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7957–7969, 2019.

[26] A. Merwaday, A. Tuncer, A. Kumbhar, and I. Guvenc, “Improved throughput coverage in natural disasters: Unmanned aerial base stations for public-safety communications,” *IEEE Veh. Technol. Mag.*, vol. 11, no. 4, pp. 53–60, 2016.

[27] Y. Lin, T. Wang, and S. Wang, “UAV-assisted emergency communications: An extended multi-armed bandit perspective,” *IEEE Commun. Lett.*, vol. 23, no. 5, pp. 938–941, 2019.

[28] N. Zhao, W. Lu, M. Sheng, Y. Chen, J. Tang, F. R. Yu, and K. Wong, “UAV-assisted emergency networks in disasters,” *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 45–51, 2019.

[29] Z. Huang, C. Chen, and M. Pan, “Multiobjective UAV path planning for emergency information collection and transmission,” *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6993–7009, 2020.

[30] 3GPP, “3GPP TR 36.777,” in *Study on Enhanced LTE Support for Aerial Vehicles (Release 15)*, Dec. 2017.

[31] Q. Liang, F. Que, and E. Modiano, “Accelerated primal-dual policy optimization for safe reinforcement learning,” *arXiv preprint arXiv:1802.06480*, 2018.

[32] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, “A Lyapunov-based approach to safe reinforcement learning,” in *Advances in neural information processing systems*, 2018, pp. 8092–8101.

[33] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.