Edge Computing Enabled Energy-Efficient Multi-UAV Cooperative Target Search

Quyuan Luo, Tom H. Luan, *Senior Member, IEEE*, Weisong Shi, *Fellow, IEEE*, and Pingzhi Fan, *Fellow, IEEE*

***Abstract*—Multiple unmanned aerial vehicles (UAVs) cooperative search have been widely adopted for surveillance and search-related applications. For a certain search area, UAVs may need to search it repeatedly to obtain a high-confidence result about the target distribution in the search area. However, the short battery life and moderate computational capability restrict UAVs to repeatedly execute the computation-intensive and energy-consuming search tasks. To address the issue, in this paper, we utilize edge computing to develop a continual and cooperative UAV search mechanism. Specifically, we first establish an edge computing enabled multi-UAV cooperative search framework, in which the mobility model of UAV, search task computing and offloading models are presented. An uncertainty minimization problem is then formulated, aiming to obtain a high-efficiency and high-confidence search result at the unpredictable uncertainty in search area. Considering that round-trip energy consumption, offloading decision-making, and trajectory planning may contribute to the reduction in uncertainty, we propose an uncertainty minimization-based cooperative target search (UMCTS) strategy. Finally, extensive simulation results validate that UMCTS can outperform the existing strategies and achieve at least 89% performance gain on average uncertainty. Based on the results, we also present a comprehensive analysis and discussion on how different parameters affect the search performance.**

***Index Terms*—Unmanned aerial vehicle, cooperative target search, edge computing, uncertainty minimization, energy-efficient offloading.**

I. Introduction

RECENTLY, utilizing multiple unmanned aerial vehicles (UAVs) to perform surveillance and search-related tasks has become a research hot spot [1], [2], [3]. For example, the UAVs can accomplish the power line inspection in a safer and more cost-efficient manner than human patrol, allowing a fast detection of a series of defects [4], [5]. Also, the UAVs can search mountainous areas after rainstorms to determine the potential mudslides, which could guide the timely evacuation and relocation of residents and important facilities that are seriously threatened.

However, due to the inevitable detection errors caused by the target detection algorithm itself, UAVs may not return absolutely credible results, resulting in *uncertainty* about the target distribution in the search area [6]. If one area is searched repeatedly, the uncertainty of this area can be reduced. A low uncertainty indicates a high reliability of search results, which helps related organizations to make better decisions based on real-time search results. However, the short battery life and moderate computational capability restrict UAVs to repeatedly execute the computation-intensive and energy-consuming search tasks [7]. To address this issue, the integration of edge computing into UAVs may significantly enhance the service capability of UAVs by offloading the intensive tasks to the edge nodes [8], making optimal offloading decisions for UAVs is a key issue for energy saving. Moreover, UAVs must decide which areas should be searched more repeatedly to reduce the uncertainty based on real-time search results. Based on the above analysis, the following two challenges must be addressed for energy saving and uncertainty minimization: *1) How to optimally make offloading decisions*, so that more energy could be saved to search more areas? *2) How to dynamically plan UAV's trajectories*, so that the uncertainty of the search area can be minimized?

Several previous efforts have focused on the edge computing enabled UAV framework [9], [10], [11], [12], [13], [14]. UAVs are mostly considered as serving nodes with powerful communication and computation capabilities to assist the mobile devices in performing some computation-intensive and latency-sensitive tasks. And the computation offloading in these works is mostly from the devices on the ground to the UAVs. However, in the multi-UAV search scenario, UAVs themselves as users would generate much data, which should be computed locally by UAVs or offloaded to the ground edge nodes. Moreover, the characteristics of the high dynamic and distributed topology of UAVs make the offloading decision-making and dynamic trajectory planning issues more challenging.

To tackle the aforementioned difficulties, in this paper, we propose an edge computing enabled multi-UAV cooperative target search strategy, where each UAV can optimally make its

Manuscript received 13 June 2022; revised 13 October 2022 and 2 December 2022; accepted 25 December 2022. Date of publication 19 January 2023; date of current version 20 June 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62101463, in part by the Natural Science Foundation of Sichuan Province of China under Grant 2022NSFSC0863, in part by the Key Research and Development Program of Sichuan under Grants 23GJHZ0209 and 23ZHSF0170, in part by the Fundamental Research Funds for the Central Universities under Grant 2682021CX044. The work of Pingzhi Fan was supported by the NSFC under Project 62020106001. The review of this article was coordinated by Prof. Zhu Han. *(Corresponding author: Quyuan Luo.)*

Quyuan Luo is with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China, also with the State Key Laboratory of Integrated Services Networks (ISN), Xidian University, Xi'an 710071, China, and also with the Provincial Key Laboratory of Information Coding and Transmission, Southwest Jiaotong University, Chengdu 611756, China (e-mail: qyluo@swjtu.edu.cn).

Pingzhi Fan is with the School of Information Science and Technology, Provincial Key Laboratory of Information Coding and Transmission, Southwest Jiaotong University, Chengdu 611756, China (e-mail: pzfan@swjtu.edu.cn).

Tom H. Luan is with the School of Cyber Engineering, Xidian University, Xi'an 710071, China (e-mail: tom.luan@xidian.edu.cn).

Weisong Shi is with the Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716 USA (e-mail: weisong@udel.edu).

Digital Object Identifier 10.1109/TVT.2023.3238040

offloading decision and dynamically plan its trajectory to minimize the uncertainty over the search area. Specifically, we first establish an edge computing enabled multi-UAV cooperative target search framework, in which the mobility model of UAV, search task computing and offloading models are presented. Then, we formulate the multi-UAV cooperative target search problem as an optimization problem to minimize the uncertainty over the search area. Considering three aspects contributing to the reduction in uncertainty, i.e., return energy consumption, offloading decision-making and trajectory planning aspects, we propose an uncertainty minimization-based cooperative target search (UMCTS) strategy. The contributions of this paper are summarized as follows.

- *Model:* Considering the imperfect target search results, we introduce the concept of *uncertainty* about the target distribution in the search area. And we establish a continual and cooperative UAV search framework, where the mobility model of UAV, search task computing and offloading models are presented.
- *Formulation:* Aiming to achieve high-efficiency and high-confidence search under limited energy of UAVs, the multi-UAV cooperative target search problem is formulated as an uncertainty minimization problem, where the offloading decisions and trajectories of UAVs are jointly optimized.
- *Algorithm:* By analyzing three aspects contributing to the reduction in energy consumption and uncertainty, we propose an uncertainty minimization-based cooperative target search (UMCTS) strategy, where return energy consumption, offloading decision-making, and trajectory planning are optimized.
- *Validation:* We compare our proposal with other schemes by simulations. The superiority of our proposal is presented. Moreover, we also conduct extensive simulations to discuss how different parameters affect the search performance under our proposal and other schemes.

The remainder of this paper is organized as follows. We present the related work in Section II. The system model and problem formulation are depicted in Section III. We elaborate on the proposed UMCTS strategy in Section IV. In Section V, performance evaluation results are presented. Finally, the conclusion is drawn in Section VI.

II. Related Works

Target search has always been a hot research topic in the field of robotics [15], [16]. In the literature, a flurry of works on target search has been reported. Many researchers focused on cooperative target search algorithms for UAV swarms/teams to complete target missions in a dynamic and risky environment. The authors in [17] formulate the cooperative target search problem as a finite horizon optimal control problem. Based on the rivaling force approach, they develop a collaborative path planning algorithm, aiming to find and confirm as many targets as possible while minimizing UAV losses. In [18], the cooperative path planning of UAVs considered in cooperative control problems involves the timing or sequencing of UAVs reaching the target or other designated locations. And the path planning problem is reduced to the problem of finding a reliable path from the initial position of a UAV to the destination. To solve the problem, probability-based algorithm, heuristic algorithm [19], grid-based search algorithm [20], and predictive algorithm [21] are mainly considered.

TABLE I
Major Notations

Notation	Explanation
$\mathcal{N}, N$	Set and number of UAVs
N_j	Number of UAVs in take-off point j
$\mathcal{G}, G$	Set and number of ground base stations
H	Altitude of UAVs
X, Y	Length and width of the search area
L_x, L_y	Number of cells along the length and width of the search area
D_{l_x,l_y}	Data size of search task T_{l_x,l_y}
C_{l_x,l_y}	Processing density of search task T_{l_x,l_y}
B	Bandwidth of each licensed channel
L	Number of channels each ground base station provides
$v_n(t)$	The position of UAV n at time step t
$x_n(t)$	The x-coordinate of position $v_n(t)$
$y_n(t)$	The y-coordinate of position $v_n(t)$
$\mathcal{O}_n^t$	Orientation set of UAV n at time step t
κ_n	Coefficient related to power UAV n
V	UAV's preset flying speed
M_n	UAV n's mass
Δ	Flying duration from one cell to its neighboring cell

With the development of intelligent systems and cooperative control theory, some work focus more on exploring intelligent cooperative target search algorithms [22], [23], [24], [25], [26]. The authors in [23] propose an intelligent self-organized algorithm (ISOA) to solve a cooperative target search planning problem for multi-UAVs. By adopting the distributed control architecture, they divide the global optimization problem into several local optimization problems. The literature [24] propose a dynamic two-stage scheme by applying the concept of the closed search to multi-UAVs cooperative target search. Similarly, to solve the closed cooperative target search problem, the authors in [25] propose an immune genetic algorithm to improve the search efficiency. Considering the complex constraints of multi-UAVs, the authors in [26] propose a dynamic discrete pigeon-inspired optimization algorithm. In a discrete environment, the proposed strategy with superior search ability can reach the global optima.

As for the literature on providing computation capabilities for computation-intensive tasks, some existing works focus on the computing offloading from mobile users to UAVs [7], [9], [27], [28], [29]. However, in some cases, the tasks generated by the UAV itself may not be able to complete. A UAV can offload computational tasks to the ground base station (GBS) supported by edge computing. Many efforts have been done on computation offloading for edge computing enabled multi-UAVs systems in recent years [7], [14], [31], [31], [32], [33], [35].

The authors in [7] design an energy-efficient computation offloading strategy for UAV-Edge computing systems. To achieve high quality of service (QoS), the authors in [31] design a UAV-Edge-Cloud computation offloading model for multi-UAVs, aiming to support computation-intensive tasks. By deploying

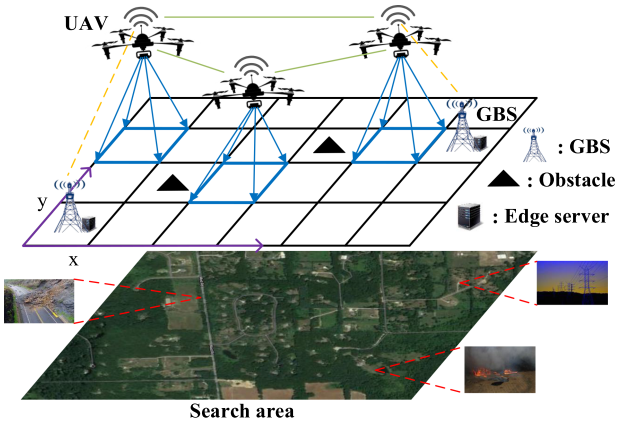


Fig. 1. Edge computing enabled multi-UAV cooperative target search framework.

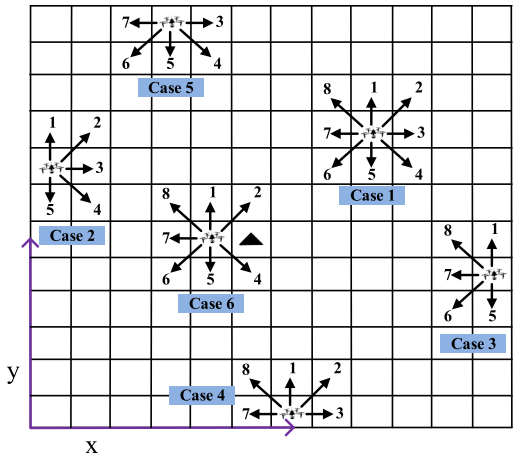


Fig. 2. Mobility model of UAVs.

edge servers on UAVs, the work in [14] studies the response delay optimization problem under communication and computation constraints. The resources at the edge of the wireless network, such as cellular GBSs, can provide cloud-like computing services to assist UAVs to complete the task processing. Cao et al. in [32] study how to offload the latency-sensitive tasks of UAVs to the GBSs, subject to the speed constraint of UAVs. Similarly, the authors in [33] studied the computation offloading problem based on two-tier UAVs, aiming to minimize the latency of tasks and the system cost.

In summary, a large amount of existing works have focused on cooperative target search or computation offloading in UAV search scenario. There has not been related work considering the inevitable detection errors and unpredictable uncertainty in search area in the multi-UAVs cooperative target search problem, which motivated this work.

III. System Model and Problem Formulation

A. System Description

For convenience, the major notations are summarized in Table I. Fig. 1 illustrates the edge computing enabled multi-UAV cooperative target search framework. We consider the search area E is a bounded $X \times Y$ area, which is further discretized and rasterized into $L_x \times L_y$ cells.[1] The UAVs fly at a constant altitude H over those cells. A bird's view camera is mounted to each UAV and can capture the video or image of a cell. Due to some high mountains or buildings, we consider there are N_t obstacles which may cause potential UAV crashes in the search area. All the hazardous cells make up the hazardous areas E_t. During task search, UAVs must avoid flying to those hazardous areas for collision avoidance. For an arbitrary hazardous area i $(1 \le i \le N_t)$, it's two-dimensional grid position is denoted by $v_i = [x_i, y_i]$ $(1 \le x_i \le L_x,\ 1 \le y_i \le L_y)$. To characterize the search task processing, we use a tuple $T_{l_x,l_y} \triangleq \{D_{l_x,l_y}, C_{l_x,l_y}\}$ to reflect the features of search task in each cell $[l_x, l_y]$, where D_{l_x,l_y} denotes the data size, and C_{l_x,l_y} denotes the processing density (in CPU cycles/bit). Since the search area is divided into multiple small cells, the features of the target search can be reflected in different data size and processing density for different cells. And N UAVs take off from N_{off} points and must return to the take-off points before they run out of energy. For an arbitrary take-off point j $(1 \le j \le N_{off})$, its grid position is denoted by $v_j = [x_j, y_j]$. The number of UAVs that would take off point j is denoted by N_j $(\sum_j N_j = N)$. Besides, G ground base stations (GBSs) are within the search area to provide powerful computing capabilities.

[1]For a given search area E, the number of cells that E can be divided into can be predefined based on the flying altitude of UAVs and the configuration of the onboard camera. In this paper, we do not focus on the division way of the search area.

B. Mobility Model of UAV

Each UAV can move from the center of its current cell to the center of one of its eight neighboring cells, subject to boundary and obstacle constraints. The position state of UAV n at time step t is denoted by $v_n(t)$, where $v_n(t) = [x_n(t), y_n(t)] \in \{1, 2, \ldots, L_x\} \times \{1, 2, \ldots, L_y\}$ refers to UAV n's position in the grid plane at time step t. Correspondingly, the real position of the UAV n at the cell with grid position $v_n(t)$ is denoted by $[(x_n(t) - 0.5)\frac{X}{L_x}, (y_n(t) - 0.5)\frac{Y}{L_y}]$. We denote the orientation set at time step t to the UAV's next position by $\mathcal{O}_n^t$, which is defined as {1 (north), 2 (northeast), 3 (east), 4 (southeast), 5 (south), 6 (southwest), 7 (west), 8 (northwest)}. As shown in Fig. 2, there are generally six cases for different orientation choices.

- *Case 1:* if none of the UAV's eight neighboring cells has boundary and obstacle constraints, the UAV can choose one orientation $o_n^t \in \{1, 2, 3, 4, 5, 6, 7, 8\}$;
- *Case 2:* if the UAV has reached the boundary of x-axis when $x_n(t) = L_x$, the UAV cannot choose one orientation $o_n^t \notin \{2, 3, 4\}$;
- *Case 3:* if the UAV has reached the boundary of x-axis when $x_n(t) = 0$, the UAV cannot choose one orientation $o_n^t \notin \{6, 7, 8\}$;
- *Case 4:* if the UAV has reached the boundary of y-axis when $y_n(t) = L_y$, the UAV cannot choose one orientation $o_n^t \notin \{1, 2, 8\}$;

- *Case 5:* if the UAV has reached the boundary of y-axis when $y_n(t) = 0$, the UAV cannot choose one orientation $o_n^t \notin \{4, 5, 6\}$;
- *Case 6:* if obstacles exist in the UAV's neighboring cells, the UAV cannot move the obstacle cells for safety (for example, the UAV cannot choose orientation 3 if an obstacle is in the eastern neighboring cell of the UAV's current position);

When UAV n moves from one cell to another, the kinetic energy consumption during flying can be expressed as

$$E_n^f(o_n^t) = 0.5M\Delta(o_n^t)V^2, \tag{1}$$

where V denotes UAV's pre-set flying speed, $\Delta(o_n^t)$ denotes flying duration, o_n^t denotes the orientation, and M denotes UAV's mass [36].

However, the flying duration $\Delta(o_n^t)$ is different when the UAV is moving from the current cell to different neighboring cells, expressed as

$$\Delta(o_n^t) = \begin{cases} \frac{Y}{L_y V}, & o_n^t = 1, 5, \\ \frac{X}{L_x V}, & o_n^t = 3, 7, \\ \frac{1}{V}\sqrt{(\frac{X}{L_x})^2 + (\frac{Y}{L_y})^2}, & o_n^t = 2, 4, 6, 8. \end{cases} \tag{2}$$

It can be drawn from formula (2) that when $o_n^t = 2, 4, 6, 8$, the flying duration has the biggest value. To achieve time-step synchronization for all UAVs, UAV's flying speed should be adjusted. To this end, we set the length of one time step as $\Delta = \frac{1}{V}\sqrt{(\frac{X}{L_x})^2 + (\frac{Y}{L_y})^2}$. Accordingly, the flying speed is correspondingly adjusted to

$$V(o_n^t) = \begin{cases} \frac{Y}{L_y \Delta}, & o_n^t = 1, 5, \\ \frac{X}{L_x \Delta}, & o_n^t = 3, 7, \\ V, & o_n^t = 2, 4, 6, 8. \end{cases} \tag{3}$$

Then the kinetic energy consumption during flying defined in the formula (1) is correspondingly re-formulated as

$$E_n^f(o_n^t) = 0.5M\Delta V(o_n^t)^2. \tag{4}$$

C. Search Task Computing and Offloading

When a UAV reached a cell, the search task can be processed locally or offloaded to the GBS. We use $\alpha_n^t = 0$ to denote that the search task is processed locally by UAV n, and $\alpha_n^t = 1$ to denote that the search task is offloaded to GBS.

1) Search Task Processed Locally: We denote the processing capability (i.e., the amount of CPU frequency in cycles/s [37]) of UAV n assigned for local computing by f_n^l. The power consumption can be modeled as

$$p_n^l = \kappa_n (f_n^l)^3, \tag{5}$$

where κ_n is a coefficient related to power [11]. And the local execution time of task T_{l_x,l_y} is then give by

$$t(l_x, l_y) = \frac{D_{l_x,l_y} C_{l_x,l_y}}{f_n^l}. \tag{6}$$

Then the energy consumption of UAV n for local processing is expressed as

$$\begin{aligned} E_n^l(l_x, l_y) &= (1 - \alpha_n^t) p_n^l t(l_x, l_y) \\ &= (1 - \alpha_n^t)\kappa_n D_{l_x,l_y} C_{l_x,l_y} (f_n^l)^2. \end{aligned} \tag{7}$$

2) Search Task Offloaded to GBS: Let L denote the number of orthogonal licensed channels each GBS provides, each with the bandwidth of B. For an arbitrary GBS $g \in \mathcal{G}$, we denote its grid position by $u_g = [x_g, y_g]$. Without loss of generality, we consider a three-dimensional (3D) Cartesian coordinate system such that the real positions of UAV n at time step t and GBS g are denoted by $[(x_n(t) - 0.5)\frac{X}{L_x}, (y_n(t) - 0.5)\frac{Y}{L_y}, H]$ and $[(x_g - 0.5)\frac{X}{L_x}, (y_g - 0.5)\frac{Y}{L_y}, 0]$, respectively. The distance between UAV n and GBS g at time step t is then given by

$$d_{n,g}(t) = \sqrt{\left((x_n(t) - x_g)\frac{X}{L_x}\right)^2 + \left((y_n(t) - y_g)\frac{Y}{L_y}\right)^2 + H^2}. \tag{8}$$

And the wireless communication data rate between UAV n and GBS g is then given by

$$R_{n,g}(t) = \frac{L}{\chi_g(t)} B \log_2\left(1 + \frac{P_n h_{n,g}(t)}{\sigma^2 d_{n,g}^{\vartheta}(t)}\right), \tag{9}$$

where P_n denotes the transmission power of UAV n, $h_{n,g}(t)$ is channel gain between UAV n and GBS g at time step t, σ^2 is the noise power, ϑ is the path loss exponent, $\chi_g(t)$ is the number of UAVs choosing to transmit their search tasks to GBS g. And all the $\chi_g(t)$ UAVs share L channels. It is noteworthy that the flying distance during task transmission is much less than the flying altitude H because the transmission time is generally very short. In this regard, for simplicity, we consider $R_{n,g}(t)$ is fixed during a certain time step. Then the energy consumption for transmitting D_{l_x,l_y} bits of search task to GBS g is expressed as

$$E_{n,g}^{tr}(t) = \frac{\alpha_n^t P_n D_{l_x,l_y}}{R_{n,g}(t)} = \frac{\alpha_n^t \chi_g(t) P_n D_{l_x,l_y}}{LB \log_2\left(1 + \frac{P_n h_{n,g}(t)}{\sigma^2 d_{n,g}^{\vartheta}(t)}\right)}. \tag{10}$$

In general, the length and width of a cell are usually tens of meters, even more than one hundred meters. The maximum UAV's speed is tens of meters per second according to the specification of DJI's UAV product [38]. And the average speed of a UAV is about a few meters per second. Accordingly, the order of magnitude of the time consumption of a UAV flying from a cell to another is usually at several or even more than ten seconds. As for the computing time, the target detection algorithm usually needs very low time consumption compared with the length of a time step. If the image size is high, the image can be compressed to a small size by many existing mature and efficient algorithms. However, since we do not focus on this point, for simplicity, we assume that processing each task would take less than one time step.

D. Problem Formulation

Since the search area E is unknown to UAVs, we consider each cell has an associated uncertainty $u(l_x, l_y, t) \in [0, 1]$, indicating UAV's uncertainty about the target distribution in that cell. $u(l_x, l_y, t) = 1$ means cell $[l_x, l_y]$ is a completely unknown area for UAVs at time step t. Uncertainty also reflects the reliability of the target detection results. The lower the uncertainty of the cell, the higher the reliability of the target detection. And $u(l_x, l_y, t)$ will be reduced as the cell is searched repeatedly, which represents less undetected information in that cell. According to the Dempster's rule of combination and the Dempster-Shafer theory (DST) [18], once a UAV has searched a cell $[l_x, l_y]$ at time step t, the uncertainty associated with that cell is reduced at an uncertainty reduction rate λ, denoted by

$$u(l_x, l_y, t+1) = \lambda u(l_x, l_y, t), \tag{11}$$

where λ can be expressed as $\lambda = 1 - \delta$, and δ is the accuracy of the target detection.

The overall objective of our system design is to find the optimal computation offloading and trajectory planning strategy for UAVs, to minimize the uncertainty over the search area under energy constraint, expressed as

$$\begin{aligned}
&\underset{\{\alpha, o\}}{\text{minimize}} \quad \mathcal{U} = \lim_{t \to \mathscr{E}} \sum_{(l_x, l_y) \in E} u(l_x, l_y, t) \\
&\text{s.t.} \quad \text{C1:} \quad \alpha_n^t \in \{0, 1\}, \\
&\qquad \text{C2:} \quad o_n^t \in \mathcal{O}_n^t, \\
&\qquad \text{C3:} \quad V(o_n^t) = \begin{cases} \frac{Y}{L_y \Delta}, & o_n^t \in \{0, 4\}, \\ \frac{X}{L_x \Delta}, & o_n^t \in \{2, 6\}, \\ V, & o_n^t \in \{1, 3, 5, 7\}, \end{cases} \\
&\qquad \text{C4:} \quad \mathscr{E} = \max_{n \in \mathbb{N}} \left\{ \mathscr{E}_n \Big| \sum_{t=0}^{\mathscr{E}_n} E_n^l(t) + E_n^{\mathsf{tr}}(t) \le \Phi_n^{\mathsf{t}=0} \right. \\
&\qquad\qquad \left. - \Phi_n^{\mathsf{ret}, \mathscr{E}_n} < \sum_{t=0}^{\mathscr{E}_n + 1} E_n^l(t) + E_n^{\mathsf{tr}}(t), \mathscr{E}_n \in N^* \right\},
\end{aligned} \tag{12}$$

where $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N]$ and $o = [o_1, o_2, \ldots, o_N]$. For arbitrary $\alpha_n \in \alpha$ and $o_n \in o$, they are sequential variables, denoted by $\alpha_n = [\alpha_n^1, \alpha_n^2, \ldots, \alpha_n^{\mathscr{E}}]$ and $o_n = [o_n^1, o_n^2, \ldots, o_n^{\mathscr{E}}]$, respectively, where $\mathscr{E}$ denotes the maximal time steps the search process can last under energy constraint. In formula (12), $t \to \mathscr{E}$ indicates that the search process will be terminated once all UAVs run out of energy. C1 indicates that a UAV can choose either local computing or offloading actions. C2 represents that UAVs can choose different orientations under boundary constraints. C3 shows the flying speed adjustment after choosing an orientation. C4 specifies the maximal time step under the energy constraint, where $\Phi_n^{\mathsf{t}=0}$ denotes the initial energy of UAV n, $\Phi_n^{\mathsf{ret}, \mathscr{E}_n}$ denotes the return energy of UAV n from cell $[x_n(\mathscr{E}_n), y_n(\mathscr{E}_n)]$ at time step $\mathscr{E}_n$ to it's take-off point j_n, expressed as

$$\Phi_n^{\mathsf{ret}, \mathscr{E}_n} = 0.5 M V d_{j_n}|_{[x_n(\mathscr{E}_n), y_n(\mathscr{E}_n)]}, \tag{13}$$

where $d_{j_n}|_{[x_n(\mathscr{E}_n), y_n(\mathscr{E}_n)]}$ is the distance from cell $[x_n(\mathscr{E}_n), y_n(\mathscr{E}_n)]$ to take-off point j_n. $\Phi_n^{\mathsf{t}=0} - \Phi_n^{\mathsf{ret}, \mathscr{E}_n}$ denotes the total energy that can can be used for local computing and transmission. The inequation in C4 indicates that a UAV has enough energy for returning to the take-off point at time step $\mathscr{E}_n$ but does not have enough energy for returning to the take-off point at next time step $\mathscr{E}_n + 1$. The purpose of C4 is to find the maximal time step under the energy constraint.

In this paper, we just consider reserving the minimum kinetic energy required for the return without considering target detection being performed by the UAV on its way back to the take-off point. This is because if the UAV performs target detection on its way back to the take-off point, more energy would be consumed thus the UAV cannot return to the take-off point. To address this issue, more energy should be reserved when the UAV makes a return decision. However, the additional energy consumption for task transmission or local computing cannot be determined when the UAV makes a return decision, resulting in that the required return energy is uncertain. Moreover, the proposed strategy for task offloading decision-making and flying orientation choosing strategy at each cell is based on the deterministic return energy at that cell. In reverse, the return energy which contains the energy for task transmission or local computing is also based on the proposed strategy. As a result, the task offloading decision-making, the flying orientation choosing, and the return energy determination are coupled. The uncertain return energy consumption when considering target detection being performed by the UAV on its way back to the take-off point would make the original problem more complex, which is hard to solve. Accordingly, in this paper, we just consider the minimum kinetic energy required for return and focus on the task offloading decision-making and the flying orientation choosing problems.

Characterized by nonconvexity and stochasticity, problem (12) is hard to solve by traditional optimization methods. To reduce the uncertainty, a UAV should try to search as many cells as possible and choose the proper cells to search. In this regard, the energy consumption should be saved as much as possible, and optimized trajectory planning is needed. For energy saving, since the obstacles are fixed, only the minimum energy consumption to make a return voyage from each cell should be maintained. Also, a UAV should optimally make its offloading decisions so that more energy could be saved to search more cells. For the optimized trajectory planning, a UAV should dynamically plan its trajectory and choose a proper orientation in each cell to reduce uncertainty. Based on the analysis above, we aim to minimize the uncertainty over the search area from the mentioned three aspects and propose a cooperative target search strategy in the following.

IV. Cooperative Target Search Strategy

A. Minimum Kinetic Energy for UAV Return

To obtain the minimum kinetic energy for UAV return, the shortest path for return should be find first based on formula (13) since UAV's mass and speed are known variables. Since the focus of this paper is not on this part, the traditional Dijkstra's

algorithm is utilized to find the shortest path from each cell to the take-off points. We regard the center point of a cell as a vertex, and the path from one cell to another as an edge. For the search area, all the vertexes of cells $\mathcal{V}$ and edges of paths $\mathcal{E}$ can form an undirected graph $< \mathcal{V}, \mathcal{E} >$. The weight of each edge is defined as the distance between the two vertexes of the edge. For the hazardous cells, we set the weights of those edges that are directly linked to an obstacle vertex to ∞. Once the undirected graph has been established, the shortest path from the take-off points to all cells can be determined based on the Dijkstra's algorithm [39]. We use $\{d_1, d_2, \ldots, d_{N_{off}}\}_{[l_x,l_y]}$ to denote the shortest distance set, where $d_j|_{[l_x,l_y]}$ $(1 \leq j \leq N_{off})$ indicates the distance from cell $[l_x, l_y]$ to take-off point j. For the return process, we consider the UAVs fly at a constant speed V. The minimum kinetic energy from cell $[l_x, l_y]$ to take-off point j is then obtained as

$$E_j^{rt}|_{[l_x,l_y]} = 0.5MVd_j|_{[l_x,l_y]}. \tag{14}$$

B. Dynamic Trajectory Planning

According to the mobility model, UAV n has a grid position $v_n(t) = [x_n(t), y_n(t)]$ and a orientation set $\mathcal{O}_n^t$ at time step t. Trajectory planning aims to decide which cell the UAV will fly to in the next time step. Each UAV uses an uncertainty map to store its knowledge about the uncertainties for all the cells and continually update it using new target detection results from its own or other UAVs by communication. And all UAVs share a common uncertainty map and update it according to formula (11). In this paper, we consider the uncertainty map sharing among UAVs is reliable and real-time through communications. This kind of uncertainty map is denoted by $\mathcal{U}(t) = \{u(l_x, l_y, t) | 0 \leq l_x \leq L_X, 0 \leq l_y \leq L_y\}$.

Aiming to minimize the uncertainty over the search area, the UAV in the current cell $[l_x, l_y]$ should fly to a neighboring cell that can reduce the uncertainty the most. Meanwhile, the UAV should have enough energy to return from that cell. To this end, the orientations the UAV can take to ensure a safe return should be first obtained. Let $\hat{\mathcal{O}}_n(t)$ denote the possible orientation set of UAV n, which is a subset of $\mathcal{O}_n^t$ (i.e., $\hat{\mathcal{O}}_n(t) \subseteq \mathcal{O}_n^t$). If an orientation $k \in \hat{\mathcal{O}}_n(t)$ is taken, the position of UAV n at time step $t+1$ will be $v_n^k = [x_n^k(t+1), y_n^k(t+1)]$. The kinetic energy consumption to v_n^k will be $E_n^f(k) = 0.5M\Delta(k)V(k)^2$. After arriving at v_n^k, the UAV executes the target search, by either processing the task locally or transmitting it to a GBS. This part of energy consumption can be expressed as $\min\{E_n^l(v_n^k), E_{n,g}^{tr}(t+1)\}$ based on formulas (7) and (10). We denote the remaining energy of UAV n before it leaves the current cell $v_n(t)$ by $E_n^{rem}(t)$, which is expressed as $E_n^{rem}(t) = E_n^{rem}(t-1) - E_n^f(o_n)|_{t-1} - min\{E_n^l(v_n(t)), E_{n,g}^{tr}(t)\}$. Consequently, all the orientations that satisfy the following energy constraint make up set $\hat{\mathcal{O}}_n(t)$:

$$E_n^{rem}(t) - E_n^f(k) - \min\{E_n^l(v_n^k), E_{n,g}^{tr}(t+1)\} \geqslant E_j^{rt}|_{v_n^k}. \tag{15}$$

Next, the orientation in $\hat{\mathcal{O}}_n(t)$ that can reduce the uncertainty the most should be chose. In formula (15), $\min\{E_n^l(v_n^k), E_{n,g}^{tr}(t+1)\}$ can determined once the offloading decision is made. We will elaborate on this part in Section IV-C.

To describe the uncertainty reduction by flying to v_n^k, we introduce a reward function to estimate the reward in v_n^k, defined as

$$\hat{\varphi}|_{v_n^k}(t+1) = u(v_n^k, t) - u(v_n^k, t+1) = (1-\lambda)u(v_n^k, t), \tag{16}$$

where $u(v_n^k, t)$ and $u(v_n^k, t+1)$ denote the uncertainty in v_n^k at time step t and $t+1$, respectively. According to (11), the uncertainty of a cell will be reduced as the cell is searched repeatedly, leading to less undetected information in that cell. If there are multiple UAVs that flying to a same v_n^k at time step $t+1$, we denote the number of those UAVs by $N|_{v_n^k}(t+1)$, then $u(v_n^k, t+1)$ can be expressed as

$$u(v_n^k, t+1) = \lambda^{N|_{v_n^k}(t+1)} u(v_n^k, t). \tag{17}$$

Thus (16) can be re-formulated as

$$\begin{aligned} \hat{\varphi}|_{v_n^k}(t+1) &= u(v_n^k, t) - u(v_n^k, t+1) \\ &= (1 - \lambda^{N|_{v_n^k}(t+1)}) u(v_n^k, t). \end{aligned} \tag{18}$$

The uncertainty over the search area at time step $t+1$ is then expressed as

$$U^{t+1} = \sum_{v_n^k \in E} u(v_n^k, t+1). \tag{19}$$

Based on (18) and (19), the total reward function of the search area by flying to v_n^k is defined as

$$\begin{aligned} \Upsilon^t &= U^t - U^{t+1} \\ &= \sum_{v_n^k \in E} u(v_n^k, t) - \sum_{v_n^k \in E} u(v_n^k, t+1) \\ &= \sum_{v_n^k \in E} (1 - \lambda^{N|_{v_n^k}(t+1)}) u(v_n^k, t). \end{aligned} \tag{20}$$

Thereafter, the original optimization problem (12) can be re-formulated as

$$\begin{aligned} &\underset{\{\alpha, o\}}{\text{maximize}} \quad \Upsilon = \sum_{t=0}^{\mathscr{E}-1} \Upsilon^t. \\ &\text{s.t. C1} \sim \text{C4} \end{aligned} \tag{21}$$

Proposition 1: The optimization problem (21) is equivalent to optimization problem (12), such that the solutions for (12) are obtained once (21) is solved.

Proof: According to the definition of Υ^t in (20), we have

$$\begin{aligned} \Upsilon &= \sum_{t=0}^{\mathscr{E}-1} \Upsilon^t \\ &= \Upsilon^0 + \Upsilon^1 + \cdots + \Upsilon^{\mathscr{E}} \\ &= (U^0 - U^1) + (U^1 - U^2) + \cdots \\ &\quad + (U^{\mathscr{E}-1} - U^{\mathscr{E}}) \end{aligned}$$

$$= U^0 - U^{\varnothing}$$
$$= U^0 - \mathcal{U}. \tag{22}$$

Since U^0 is a constant, to maximize Υ in problem (21) is equivalent to minimize $\mathcal{U}$ in problem (12). Accordingly, optimization problem (21) is equivalent to (12). *Proposition 1 is proved.* ■

Solving problem (21) can be approximately equivalent to obtain the maximal Υ^t at each time step. Obviously, Υ^t is an increasing function of $u(v_n^k, t)$. And a UAV should fly to a v_n^k that has the highest $u(v_n^k, t)$ to obtain a highest reward. In this paper, we assume that UAVs would fly at slightly different altitudes in order to minimize the risk of collisions if multiple UAVs fly to a cell at the same time. And this can be achieved through the wireless communications between adjacent UAVs. In this case, the reward obtained by UAV n would be

$$\varphi|_{v_n^k}(t+1) = \frac{1}{N|_{v_n^k}(t+1)}\left\{u(v_n^k, t) - u(v_n^k, t+1)\right\}$$
$$= \frac{1-\lambda^{N|_{v_n^k}(t+1)}}{N|_{v_n^k}(t+1)} u(v_n^k, t). \tag{23}$$

Since $\lambda = 1-\delta$, $\varphi|_{v_n^k}(t+1)$ is an increasing function of $\delta|_{t+1}$. Now, we need to get the value of $N|_{v_n^k}(t+1)$. Let $\mathcal{N}_{\{v_n^k\}}$ denote the set of UAVs that can fly to cell v_n^k at time step $t+1$, and for an arbitrary UAV $m \in \mathcal{N}_{\{v_n^k\}}$, its possible orientation set at time step t is denoted by $\hat{\mathcal{O}}_m(t)$. Let $v_m^l = [x_m^l(t+1), y_m^l(t+1)]$ denote the cell position at time step $t+1$ after orientation $l \in \hat{\mathcal{O}}_m(t)$ is taken by UAV m and $u(v_m^l, t)$ denote the uncertainty of cell v_m^l at time t. Observing that $\varphi|_{v_n^k}(t+1)$ is a decreasing function of $N|_{v_n^k}(t+1)$, we design a reward-based orientation determination (ROD) algorithm to determine the value of $N|_{v_n^k}(t+1)$ and whether a UAV would fly to cell v_n^k, which is described in the following five steps and presented in Algorithm 1.

1) For each $m \in \mathcal{N}_{\{v_n^k\}}$, judge whether the uncertainty of v_n^k is the largest among $\{v_m^l | l \in \hat{\mathcal{O}}_m(t)\}$, if not, delete m from $\mathcal{N}_{\{v_n^k\}}$;
2) If $\mathcal{N}_{\{v_n^k\}} \neq \varnothing$, for each m, calculate $\hat{\varphi}|_{v_m^l}(t+1)$ according to formula (16) over each l, and store all $\hat{\varphi}|_{v_m^l}(t+1)$ values in set $\{\hat{\varphi}|_{v_m^l}(t+1)\}_{l\in\hat{\mathcal{O}}_m(t)}$. And the second largest value (if exists) in $\{\hat{\varphi}|_{v_m^l}(t+1)\}_{l\in\hat{\mathcal{O}}_m(t)}$ is denoted as $\hat{\varphi}|_m^{2nd}$;
3) Let $N|_{v_n^k}(t+1) = |\mathcal{N}_{\{v_n^k\}}|$, calculate $\varphi|_{v_n^k}(t+1)$ according to formula (23);
4) Among all the $m \in \mathcal{N}_{\{v_n^k\}}$ that meet $\hat{\varphi}|_m^{2nd} > \varphi|_{v_n^k}(t+1)$, delete the m with the largest $\hat{\varphi}|_m^{2nd}$ from $\mathcal{N}_{\{v_n^k\}}$. And the UAV m should fly to the cell with the reward value of $\hat{\varphi}|_m^{2nd}$;
5) Go back to step 3) and step 4), until there is no UAV in $\mathcal{N}_{\{v_n^k\}}$ that meets $\hat{\varphi}|_m^{2nd} > \varphi|_{v_n^k}(t+1)$.

To make the ROD algorithm easier to understand, we present the following case as an example. Assume there are four UAVs (denoted by $U1$, $U2$, $U3$, and $U4$) that can fly to a common cell at time step $t+1$. The uncertainty cell sets the four UAVs can fly to are $\{0.8, 0.7, 0.6\}|_{U1}$, $\{0.2, 0.3, 0.6\}|_{U2}$, $\{0.3, 0.44, 0.6\}|_{U3}$, $\{0.4, 0.5, 0.6\}|_{U4}$, respectively. In this case, the uncertainty of the common cell v_n^k is 0.6. At first, $\mathcal{N}_{\{v_n^k\}} = \{U1, U2, U3, U4\}$. Since 0.6 is not the largest uncertainty among $\{0.8, 0.7, 0.6\}|_{U1}$, delete $U1$ from $\mathcal{N}_{\{v_n^k\}}$. Now, $\mathcal{N}_{\{v_n^k\}} = \{U2, U3, U4\}$. We assume $\lambda = 0.5$, after calculating $\hat{\varphi}|_{v_m^l}(t+1)$, the reward sets of the three UAVs in $\mathcal{N}_{\{v_n^k\}}$ at time step $t+1$ are $\{0.1, 0.15, 0.3\}|_{U2}$, $\{0.15, 0.22, 0.3\}|_{U3}$, and $\{0.2, 0.25, 0.3\}|_{U4}$, respectively. Now, $\hat{\varphi}|_{U2}^{2nd}$, $\hat{\varphi}|_{U3}^{2nd}$, and $\hat{\varphi}|_{U4}^{2nd}$ are 0.15, 0.22, and 0.25, respectively, and $\varphi|_{v_n^k}(t+1) = \frac{1-0.5^3}{3} \times 0.6 = 0.175$. Since $\hat{\varphi}|_{U3}^{2nd} = 0.22$ and $\hat{\varphi}|_{U4}^{2nd} = 0.25$ are both bigger than 0.175, and $\hat{\varphi}|_{U4}^{2nd} > \hat{\varphi}|_{U3}^{2nd}$, delete $U4$ from $\mathcal{N}_{\{v_n^k\}}$, thus $U4$ should fly to the cell with the reward value of $\hat{\varphi}|_{U4}^{2nd} = 0.25$. Now, $\mathcal{N}_{\{v_n^k\}} = \{U2, U3\}$, and $\varphi|_{v_n^k}(t+1) = \frac{1-0.5^2}{2} \times 0.6 = 0.225$. Since neither $\hat{\varphi}|_{U2}^{2nd}$ or $\hat{\varphi}|_{U3}^{2nd}$ is bigger than 0.225, both $U2$ and $U3$ would fly to v_n^k.

Algorithm 1: Reward-Based Orientation Determination (ROD) Algorithm.

```
Obtain N_{v_n^k}
for each m ∈ N_{v_n^k}:
  if ∃l ∈ Ô_m(t), u(v_m^l, t) > u(v_n^k, t) then
    Delete m from N_{v_n^k}
  end if
end for
if N_{v_n^k} ≠ ∅ then
  for each m ∈ N_{v_n^k}:
    for each l ∈ Ô_m(t):
      Calculate φ̂|_{v_m^l}(t+1) based on formula (16)
      and store it in set {φ̂|_{v_m^l}(t+1)}_{l∈Ô_m(t)}
    end for
  end for
end if
if |{φ̂|_{v_m^l}(t+1)}_{l∈Ô_m(t)}| > 1 then
  Denote the second largest value in
  {φ̂|_{v_m^l}(t+1)}_{l∈Ô_m(t)} by φ̂|_m^{2nd}
end if
while ∃m ∈ N_{v_n^k}, φ̂|_m^{2nd} > φ|_{v_n^k}(t+1) :
  Let N|_{v_n^k}(t+1) = |N_{v_n^k}|, calculate φ|_{v_n^k}(t+1)
  based on formula (23)
  for each m ∈ N_{v_n^k}:
    if φ̂|_m^{2nd} > φ|_{v_n^k}(t+1) then:
      Delete m from N_{v_n^k}
      UAV m should fly to the cell with the reward
      value of φ̂|_m^{2nd}
    end if
  end for
end while
```

C. Energy-Efficient Offloading Decision Making

To minimize the value of $\min\{E_n^l(v_n^k), E_{n,g}^{tr}(t+1)\}$ in formula (15), we design an energy-efficient offloading decision making strategy. For convenience, we use E^{pro} to denote $\min\{E_n^l(v_n^k), E_{n,g}^{tr}(t+1)\}$. Once a UAV flies to a certain cell, the value of $E_n^l(v_n^k)$ is fixed according to formula (7), and E^{pro} can be determined if the value of $E_{n,g}^{tr}(t+1)$ can be

obtained. As formulated in formula (10), $E^{tr}_{n,g}(t+1)$ is an increasing function of $\chi_g(t+1)$ and $d_{n,g}(t+1)$. To minimize $E^{tr}_{n,g}(t+1)$, a UAV should choose a GBS with a smaller $\chi_g(t+1)$ and $d_{n,g}(t+1)$. However, for a certain UAV n, how to know the number of UAVs that offload their tasks to a certain GBS is a challenging issue. To address this issue, we design a potential UAV based energy-efficient offloading decision-making (PUEOD) algorithm, which is introduced in the following.

During the uncertainty map sharing process, we consider that a UAV also broadcasts its position such that UAV n has the position information of other UAVs.[2] Combining the positions of all GBSs and UAVs at time step t, the number of UAVs within the coverage of a certain GBS at time step $t+1$ can be determined. Here, we introduce the concept of *potential UAVs* within a certain GBS. A UAV can be regarded as a potential UAV within a certain GBS only if it may arrive at a position that is within the coverage of the GBS at time step $t+1$. The concept of potential UAVs is used to express $\chi_g(t+1)$. Accordingly, if UAV n choose to arrive at v^k_n, which is assumed to be within the coverage of GBS g, the number of potential UAVs within g can be determined. Let $\mathcal{C}_g$ denote the coverage cell set of GBS g, and $\chi_g(t+1)$ can be expressed as

$$\chi_g(t+1) = \sum_{i \in \mathcal{N} \setminus \{n\}} \mathbf{1}\left\{\exists k \in \mathcal{O}_i(t), v^k_i \in \mathcal{C}_g\right\} + 1, \quad (24)$$

where $\mathbf{1}\{\tau\}$ is an indicator function which equals 1 if τ is true and 0 otherwise.

Once $\chi_g(t+1)$ is obtained, $E^{tr}_{n,g}(t+1)$ can be obtained through replacing $\chi_g(t)$ with $\chi_g(t+1)$ in formula (10). For each GBS $g \in \mathcal{G}$, UAV n can obtain the transmission energy value $E^{tr}_{n,g}(t+1)$. Among all the values, the minimum (denoted by $E^{tr}_{n,\min}$) is regarded as the transmission energy value if UAV n would offload its task to that GBS. Finally, the minimal transmission energy is compared with the local computing energy. If $E^l_n(v^k_n) \le E^{tr}_{n,\min}$, UAV n would process the task locally, and formula $\min\{E^l_n(v^k_n), E^{tr}_{n,g}(t+1)\} = E^l_n(v^k_n)$ holds; otherwise, UAV n would transmit it to g, and formula $\min\{E^l_n(v^k_n), E^{tr}_{n,g}(t+1)\} = E^{tr}_{n,\min}$ holds.

By integrating the processes of the return energy calculation, offloading decision making, and orientation determination, we propose an uncertainty minimization-based cooperative target search (UMCTS) strategy, which is presented in Algorithm 2. The UMCTS strategy mainly includes three parts. The first part is used for calculating the minimum kinetic energy for UAV return, as shown in Lines 6-8 of Algorithm 2. The second part is used for making energy-efficient offloading decisions, as shown in Line 13-14 of Algorithm 2. The third part is used for choosing optimal flying orientations, as shown in Line 21 of Algorithm 2.

[2]Actually, UAVs can share information with other UAVs through the star or mesh UAV network [25], the coalition-based UAV network [40], and the optimized LTE network for UAV communications [41], [42]. For simplicity, in this paper, we do not focus on the communication structure of UAV network, and just assume UAVs can share uncertainty map and position with other UAVs.

Algorithm 2: Uncertainty Minimization-Based Cooperative Target Search (UMCTS) Strategy.

1: Initialize search area E, hazardous area E_t
2: Initialize N UAVs, N_{off} take-off points
3: Initialize $E^l_n(0) = 0$, $E^{tr}_{n,g}(0) = 0$
4: **for** each cell $[l_x, l_y]$ ($[l_x, l_y] \notin E_t$) in parallel:
5: Initialize uncertainty $u([l_x, l_y], 1)$
6: Obtain the undirected graph $< \mathcal{V}, \mathcal{E} >$
7: Calculate the shortest distance set $\{d_1, d_2, \ldots, d_{N_{off}}\}_{[l_x,l_y]}$
8: Obtain the minimum energy $\{E^{rt}_1, E^{rt}_2, \ldots, E^{rt}_{N_{off}}\}|_{[l_x,l_y]}$ based on formula (14)
9: **for** time step $t = 0, 1 \ldots$:
10: **for** UAV $n \in N$ in parallel:
11: **do:**
12: Obtain position set $\mathcal{O}^t_n$
13: Execute the potential UAV based energy-efficient offloading decision-making (PUEOD) algorithm
14: Make the optimal offloading decision and obtain $\min\{E^l_n(v^k_n), E^{tr}_{n,g}(t+1)\}$
15: According to the formula (17), obtain $\hat{\mathcal{O}}_n(t)$
16: **if** $\hat{\mathcal{O}}_n(t) = \varnothing$ **then**
17: UAV n returns to take-off point based on the obtained shortest path
18: **else**
19: Obtain the common cells that multiple UAVs can fly to according to all $\hat{\mathcal{O}}_n(t)$
20: **if** exists common cells **then**
21: Execute the reward-based orientation determination (ROD) algorithm to choose the optimal orientation
22: **else**
23: Choose a orientation from $\hat{\mathcal{O}}_n(t)$ whose position has the highest uncertainty
24: **end if**
25: **end if**
26: **until** all UAVs return to the take-off points
27: **end for**
28: **end for**

D. Analysis on Complexity, Optimality and Convergence of UMCTS

The complexity of UMCTS mainly comes from three aspects, corresponding to the three parts of UMCTS. The first aspect is from the computation of the minimum kinetic energy for UAV return. As presented in Section IV-A, the complexity is mainly from the Dijkstra's algorithm, which can be calculated as $O(N^2_{cell})$, where N_{cell} denotes the number of cells. The second aspect is from the potential UAV based energy-efficient offloading decision-making (PUEOD) algorithm, as presented in Section IV-C. In PUEOD, the complexity is mainly from the calculation of potential UAVs in each GBS. The number of iterations in the outer and inner loops are the number of GBSs G and the number of UAVs N respectively, the complexity can

be calculated as $O(GN)$. The third aspect is from the reward-based orientation determination (ROD) algorithm, as presented in Algorithm 1 of Section IV-B. In ROD, the complexity is mainly from the Lines 2-6 for judging the largest uncertainty cell, Lines 7-13 for calculating the reward function, and Lines 17-25 for determining the optimal orientations of UAVs. Suppose the number of UAVs that can fly to a common cell is N_c, N_o denotes the number of possible orientations of a UAV in the common cell. The complexity of ROD can be calculated as $O(N_c) + O(N_cN_o) + O(N_c!)$ in the worst case and $O(N_c) + O(N_cN_o)$ in the best case. Since the values of N_c and N_o are generally very small. The complexity of ROD is generally very low. Thus, the total complexity of the proposed UVMCTS can be expressed as $O(N_{cell}^2) + O(GN) + O(N_c) + O(N_cN_o) + O(N_c!)$ in the worst case, and $O(N_{cell}^2) + O(GN) + O(N_c) + O(N_cN_o)$ in the best case.

As for the optimality of UMCTS, by introducing a reward function, the uncertainy minimization problem (12) is transformed to a reward maximization problem (21). (21) is proved equivalent to (12), such that the solutions for (12) are obtained once (21) is solved, which has been verified in Proposition 1 and it's corresponding proof. The Dijkstra's algorithm based minimum kinetic energy calculation and the potential UAV based energy-efficient offloading decision-making both can guarantee that UAV's energy can be maximally used for the search process to maximize the reward. In the design of ROD algorithm, we not only record the largest reward value of the possible cells, but also the second largest one. This operation can avoid the situation when several UAVs fly to a cell to pursuit the best reward, however a second-best reward is obtained for each UAV. For such UAVs, flying to a neighboring cell with the second largest reward may obtain an optimum. The ROD algorithm can guarantee that all UAVs can fly to their corresponding cells to obtain a maximum reward such that the uncertainty of the search area is minimized at each time step. Considering all possible future time steps at the current time step to make trajectory planning and offloading decision may lead to a global optimum theoretically. However, it will make the optimization problem difficult to solve or even unsolvable, which may result in a very high complexity. Although only an approximated optimization solution to (21) is obtained by finding the maximal Υ^t at each time step, the proposed UMCTS runs in a polynomial time [14].

As for the convergence of UMCTS, since Υ^t is an increasing function of $u(v_n^k, t)$, a UAV should fly to a v_n^k that has the highest $u(v_n^k, t)$ to obtain a highest reward. Moreover, the ROD algorithm guarantees an optimal reward for each UAV. The accumulated reward is upper bounded by the situation when all cells are uniformly searched such that the uncertainty is lower bounded by a corresponding threshold. Accordingly, the proposed UMCTS will finally converge to an optimum after a finite number of time steps when UAVs run out of energy.

V. Performance Evaluation

A. Experimental Setup

We consider a 400 m $\times$ 400 m search area, which is further discretized and rasterized into 20 $\times$ 20 cells. Four GBSs are deployed at four corners of the search area[3] UAVs take off at the four corners (A1, A2, A3, and A4), as illustrated in Fig. 3. Other parameters setting of UAVs is listed in Table II. For the dataset, we use the process of license plate recognition (LPR) to simulate the image recognition of the target search. And we adopt the HyperLPR project from GITHUB [43] and choose 10000 images from the Chinese City Parking Dataset (CCPD) [44]. We use a GPU-based server with four NVIDIA GTX 2080 Ti GPUs, where the CPU is Intel Xeno E5-2690 v4 with 64-GB memory. For the software environment, we deploy Docker Containers [45] in the server to simulate the actions and interactions of GBSs and UAVs. When a UAV flies to a cell, an image is randomly chosen among the 10000 images and is processed by UAV or GBS.

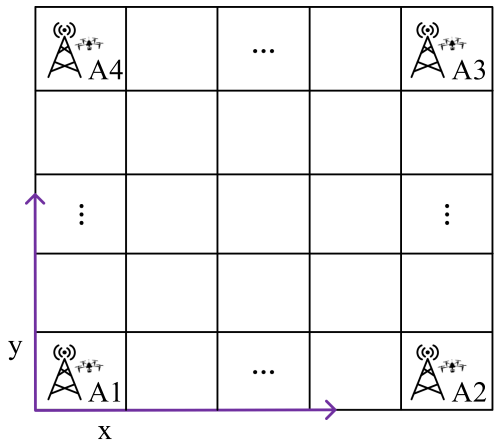


Fig. 3. Simulation scenario.

TABLE II
Main Parameters Setting

Description	Value
Altitude of UAVs (H)	20 m
Bandwidth per channel (B)	$0.1 \sim 0.5$ MHz
Channel power gain (h_0)	-50 dB
Mass of UAVs (M)	$0.5 \sim 3$ kg
Number of channels (L)	4
Number of UAVs (N)	$1 \sim 4$
Processing density of images (C_{l_x,l_y})	2000 cycles/bit
Coverage radius of GBS	400 m
Pre-set flying speed of UAVs (V)	$5 \sim 20$ m/s
Processing capability of UAVs (f_n^l)	1×10^9
Switched capacitance coefficient (k_n)	1×10^{-24}
Transmission power of UAVs (P_n)	20 dBm
White Gaussian noise power (σ_0^2)	1×10^{-9} W

To verify the performance of our proposed UMCTS, we introduce the following three policies as benchmarks:

- *Local-Comp-Only* (LCO), where all UAVs compute their computation tasks locally;
- *Offload-Comp-Only* (OCO), where all UAVs offload their computation tasks to GBSs to process;
- *Random-Offload* (RO), where each UAV chooses local computing or offloading randomly;

[3]Although one GBS can be placed in the center and cover the whole area, the optimal GBS choosing process cannot be validated by the proposed PUEOD. Therefore, 4 GBSs are considered.

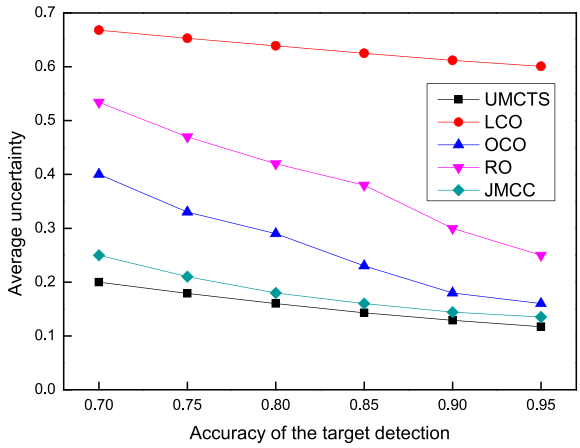
- *Non-Coop-Search* (NCS), where UAVs search the target area in a non-cooperative manner.
- *Joint Mobility, Communication and Computation based scheme (JMCC)* [46], where the mobility, communication and computation optimization are jointly considered while the uncertainty minimization is ignored.

It is noteworthy that all UAVs in LCO, OCO, RO, and JMCC share a common uncertainty map in a cooperative manner, while each UAV in NCS maintains a private uncertainty map without sharing it with other UAVs in a non-cooperative manner.

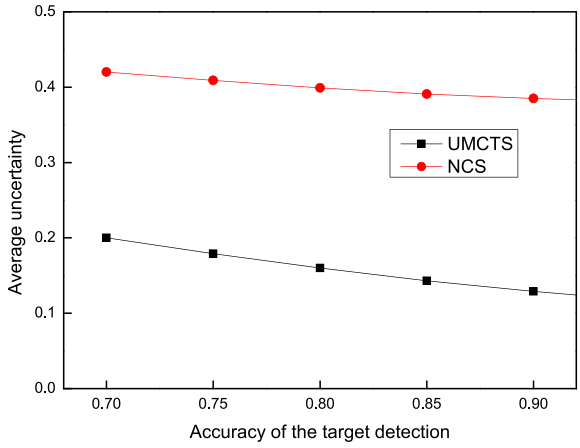
B. Simulation Results

1) Effectiveness: In this set of simulations, we set $N = 2$ and two UAVs take off from A1 and A3, respectively. We randomly choose 15 hazardous cells over the search area. The initial energy of each UAV is set to 2×10^5 Joule,[4] and the mass of each UAV is set to 1 kg. Fig. 4 shows the relationship between the average uncertainty of all cells and the accuracy of the target detection σ under different schemes. Especially, Fig. 4(a) illustrates the performance comparison between UMCTS and four cooperative schemes, i.e., LCO, OCO, RO, and JMCC. Fig. 4(b) illustrates the performance comparison between UMCTS and the non-cooperative scheme NCS. It is obviously from Fig. 4(a) that the average uncertainty of the search area decreases with the increasing accuracy of the target detection. Because the higher the accuracy of the target detection, the easier it is to identify the search area, thus leading to a lower uncertainty of the search area, which is also consistent with formula (11). Moreover, the average uncertainty of UMCTS is lower than that of LCO, OCO, RO, and JMCC, which can reduce the average uncertainty by up to 80%, 50%, 62%, and 20% over the four schemes, respectively. This is because the UAVs in UMCTS not only execute a ROD algorithm to choose an optimal flying orientation for reducing uncertainty the most, but also execute the PUEOD algorithm to optimally choose local computing or offloading for energy saving. However, more computing energy or more transmission energy will be consumed if the PUEOD algorithm is not considered in LCO, OCO, and RO. Accordingly, fewer area would be searched, leading to higher uncertainty. In JMCC, although the joint mobility, communication, and computation optimization are considered, the second largest reward orientation is ignored, resulting in the situation when several UAVs fly to a cell to pursuit the best reward, however a second-best reward is obtained for each UAV. Accordingly, the search performance of JMCC is inferior to our proposed UMCTS. Also, it can be seen from Fig. 4(a) that offloading tasks to GBS is more beneficial to uncertainty reduction than local computing, which indicates that the offloading operation would consume less energy than local computing in most cases under this simulation scenario.

Also, the performance comparison between UMCTS and NCS is shown in Fig. 4(b). Here, the average uncertainty of NCS



(a)



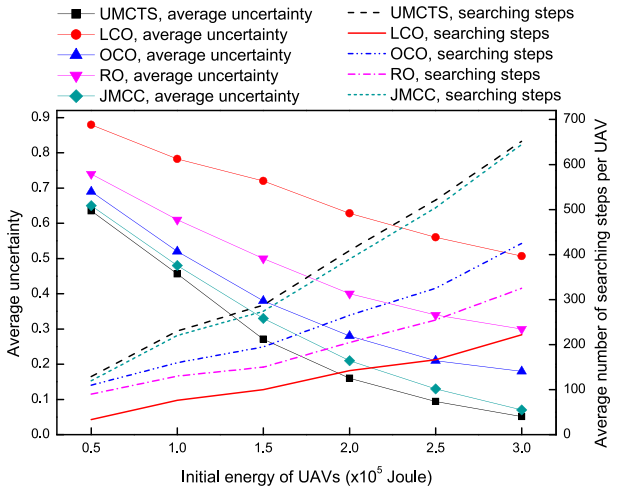
(b)

Fig. 4. Average uncertainty of search area under different accuracy of target detection. (a) Comparison of average uncertainty with cooperative schemes. (b) Comparison of average uncertainty with non-cooperative scheme.

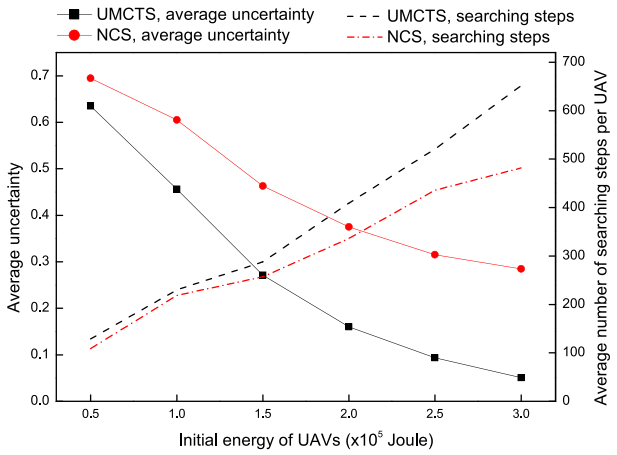
is obtained by averaging all UAVs' uncertainty map values when all UAVs finish their search tasks. It can be seen that UMCTS outperforms NCS and can reduce the average uncertainty by up to 69% over NCS. This is because, during the search process, the UAVs in the NCS do not share their uncertainty maps, leading to a repeated search of a cell that has been already searched by other UAVs, ignoring those cells with higher uncertainty. However, according to formula (16), a UAV should fly to a cell with the highest uncertainty to obtain a highest reward such that reducing the uncertainty the most. Therefore, UMCTS outperforms the NCS. Especially, when the target cells are searched repeatedly by UAVs, the superiority of UMCTS over the non-cooperative scheme is more prominent. To verify this point, we will further show the performance gap between UMCTS and NCS under different initial energy of UAVs in Fig. 5(b).

2) Effect of Initial Energy of UAVs on Search Performance: In this set of simulations, we set accuracy $\delta = 0.8$. Let the initial energy of UAVs vary from 0.5×10^5 Joule to 3.0×10^5 Joule, Fig. 5 shows the relationship between average uncertainty and

[4]According to the products of DJI [38] whose voltage is about 15 V and the conversion from mAh to Joule based on voltage 15 V, i.e., 1 mAh $= 0.001\ A \times 15\ V \times 3600\ s = 54 Joule$, the setting of $0.5 \times 10^5 \sim 3 \times 10^5 Joule$ means $926 \sim 5555 mAh$.



(a)



(b)

Fig. 5. Average uncertainty of search area under different initial energy of UAVs. (a) Comparison with cooperative schemes. (b) Comparison with non-cooperative scheme.

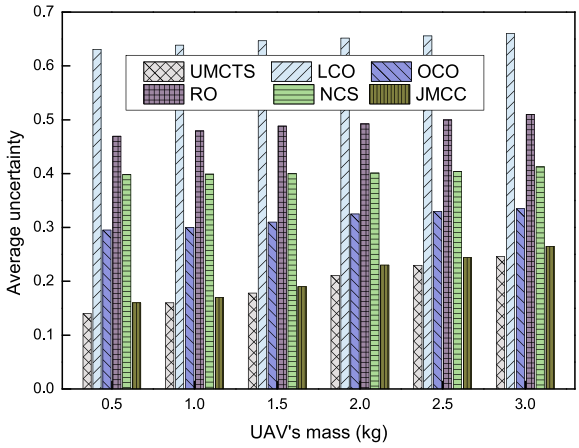
initial energy. For the comparison with cooperative schemes LCO, OCO, RO, and JMCC Fig. 5(a) shows that the average uncertainty of all five schemes decreases with the increasing initial energy. This is because a UAV with more energy can fly to more cells to execute search tasks, leading to a reduction in uncertainty. The right Y-axis of Fig. 5 exactly shows that the average number of searching steps per UAV (which also means the average number of cells searched by each UAV) increases with the increasing initial energy. Obviously, UMCTS achieves the best search performance on both the average number of searching steps and average uncertainty among the four cooperative schemes. Specifically, UMCTS can improve the average number of searching steps by up to 285%, 60%, 104%, and 8% and reduce the average uncertainty by up to 89%, 72%, 83%, 23% and 28% over LCO, OCO, RO, and JMCC, respectively.

Also, for the comparison with the non-cooperative scheme, as shown in Fig. 5(b). It is obvious that the average uncertainty of both UMCTS and NCS also decreases with the increasing initial energy of UAVs. It is noteworthy that the performance gap between UMCTS and NCS becomes bigger when the initial energy of UAVs increases. And UMCTS can improve the average number of searching steps by up to 35%, and reduce the average uncertainty by up to 82% over NCS. This is because when the initial energy of UAVs is small, the number of target cells a UAV can search is also small, resulting in a small probability that some target cells are searched repeatedly by multiple UAVs. With the increase of energy, UAVs can search for more target cells. UAVs in UMCTS would fly to a cell that has the highest uncertainty to reduce the uncertainty the most through the shared uncertainty map. However, the UAVs in the NCS may repeatedly search the cells that have been already searched by other UAVs, ignoring those cells with higher uncertainty. Moreover, It can be also drawn from both Fig. 5(a) and (b) that to achieve a same uncertainty, the UAVs in our proposed UMCTS just need less initial energy. This further verifies the energy-efficient characteristic of our proposed UMCTS.

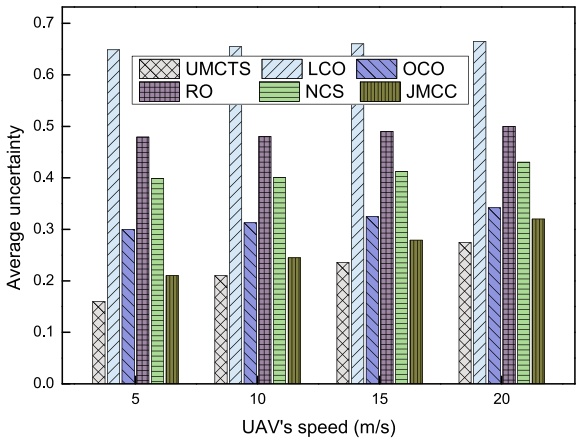
3) Effect of UAV's Mass on Search Performance: According to formula (1), the kinetic energy consumption during flying is proportional to UAV's mass. To reveal the effect of UAV's mass on search performance, let the UAV's mass vary from 0.5 kg to 3 kg. Fig. 6(a) illustrates how UAV's mass affects the search performance. Obviously, the search performance degrades with the increasing UAV's mass for all six schemes. And UMCTS has the best search performance among them, which can reduce the average uncertainty by up to 78%, 47%, 70%, 65%, and 13% over LCO, OCO, RO, NCS, and JMCC respectively. It is noteworthy that when UAV's mass is reduced gradually from 3 kg to 0.5 kg, there is no significant improvement on average uncertainty of LCO, RO, OCO, and NCS compared with UMCTS. This is because LCO, RO, OCO, and NCS have fewer searching steps than UMCTS as presented in Fig. 5, and less kinetic energy will be consumed. Consequently, no significant change on average uncertainty would be produced when changing UAV's mass. More importantly, it can be concluded that given the same condition, the UAV's mass in UMCTS should be reduced as much as possible to get a better search performance.

4) Effect of UAV's Speed on Search Performance: In addition to UAV's mass, according to formula (1) and $E_j^{rt}|_{v_n^k} = 0.5MVd_j|_{v_n^k}$, UAV's flying speed also affects kinetic energy consumption. Let the UAV's speed vary from 5 m/s to 20 m/s, Fig. 6(b) illustrates how UAV's speed affects the search performance. It can be seen that the average uncertainty increases with the increasing UAV's speed. And UMCTS outperforms LCO, OCO, RO, NCS, and JMCC which can reduce the average uncertainty by up to 75%, 47%, 67%, 60%, and 23% over the five schemes. However, there is only a small search performance deterioration of LCO, RO, OCO, and NCS when increasing the UAV's speed compared with UMCTS. This is also because LCO, RO, OCO, and NCS have fewer searching steps than UMCTS, the increase of kinetic energy consumption caused by the increase of speed has little influence on the search performance. More importantly, it can be concluded that given the same condition and if time permits, reducing UAV's speed could obtain a better search performance.

5) Effect of Bandwidth on Search Performance: Let the bandwidth of each wireless communication channel vary from

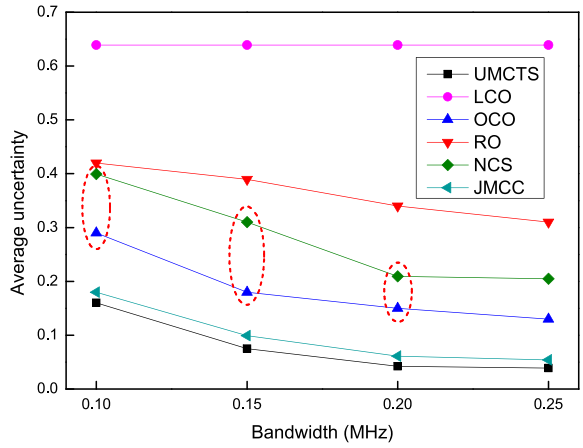


(a)

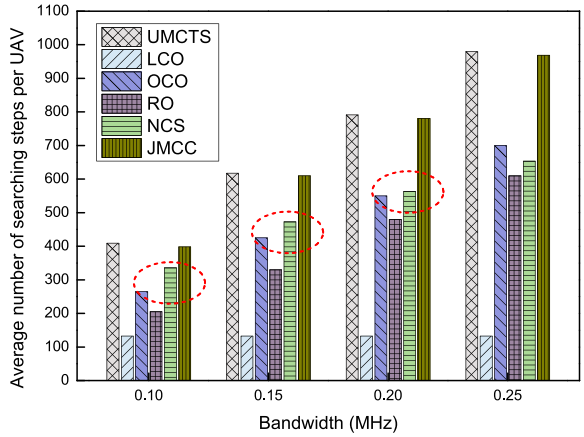


(b)

Fig. 6. Average uncertainty under different UAV's mass and speed. (a) Average uncertainty under different UAV's mass. (b) Average uncertainty under different UAV's speed.



(a)

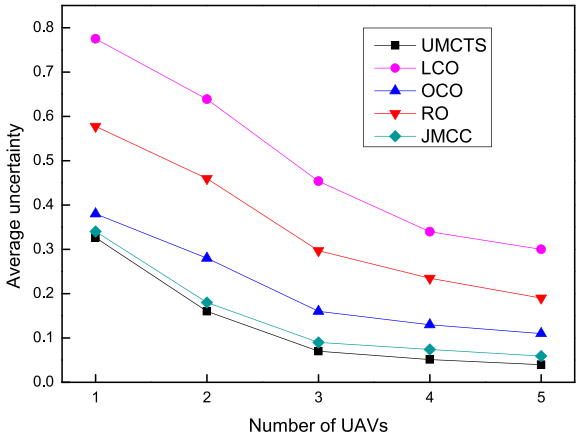


(b)

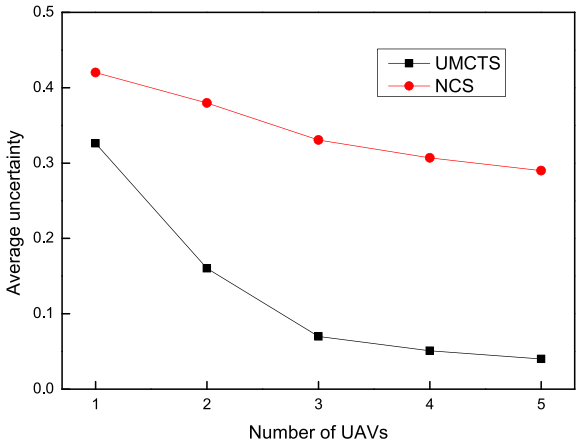
Fig. 7. Search performance under different bandwidth. (a) Average uncertainty under different bandwidth. (b) Average number of searching steps under different bandwidth.

0.1 MHz to 0.25 MHz. Fig. 7(a) shows that the average uncertainty of UMCTS, OCO, RO, NCS and JMCC decreases with the increasing bandwidth while that of LCO keeps unchanged. This is because there is no offloading process in LCO. Therefore, the search performance has nothing to do with bandwidth. This point of view can be verified in Fig. 7(b) because the average number of searching steps of LCO keeps unchanged with different bandwidths while that of UMCTS, OCO, RO, and NCS increase with the increasing bandwidth. It is noteworthy that the average number of searching steps of NCS is larger than that of OCO as the red circled part in Fig. 7(b). However, the search performance of NCS is worse than that of OCO. This is because a UAV in NCS may fly to a cell without the highest uncertainty due to the non-cooperative manner. From this point of view, the superiority of UMCTS is further verified in Fig. 7. Specifically, UMCTS can improve the average number of searching steps by up to 640%, 54%, 99%, 65%, and 28% and reduce the average uncertainty by up to 94%, 72%, 88%, 81%, and 4% over LCO, OCO, RO, NCS, and JMCC, respectively.

6) Effect of Number of UAVs and Cells on Search Performance: Let the number of UAVs vary from 1 to 5. Specifically, when $N = 1$, a take-off cell from {A1, A2, A3, A4} is randomly chose; when $N = 2$, {A1, A3} or {A2, A4} is randomly chose; when $N = 3$, a cell from {A1, A2, A3, A4} is randomly chose to be not as the take-off cell; when $N = 4$, four UAVs take off at {A1, A2, A3, A4} respectively; and when $N = 5$, a cell from {A1, A2, A3, A4} is randomly chosen as take-off cell for random two UAVs. Both Fig. 8(a) and (b) show that the search performance is improved when more UAVs participate in the search process. And UMCTS outperforms the benchmarks, which can reduce the average uncertainty by up to 87%, 64%, 79%, 86%, and 25% over LCO, OCO, RO, NCS, and JMCC, respectively. It is noteworthy that the performance gap between UMCTS and NCS becomes bigger when the number of UAVs increases. This is because more UAVs in UMCTS would fly to cells with higher uncertainty to reduce the uncertainty through the shared uncertainty map. More importantly, when the number of UAVs increases from 3 to 5, just a small performance improvement is obtained. This is because when $N = 3$, the average



(a)



(b)

Fig. 8. Search performance under different number of UAVs. (a) Comparison with cooperative schemes. (b) Comparison with non-cooperative scheme.

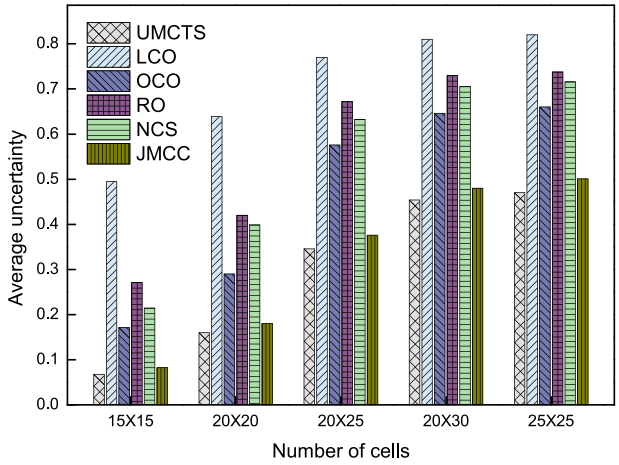
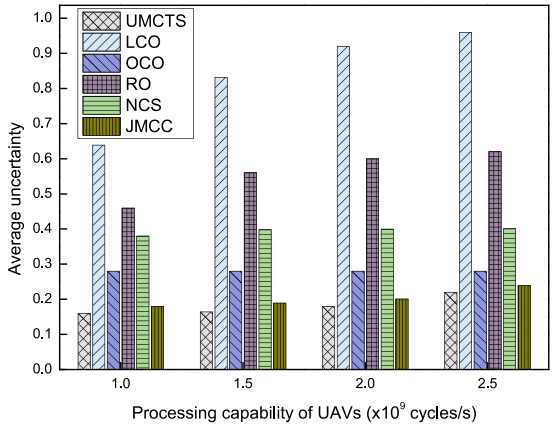
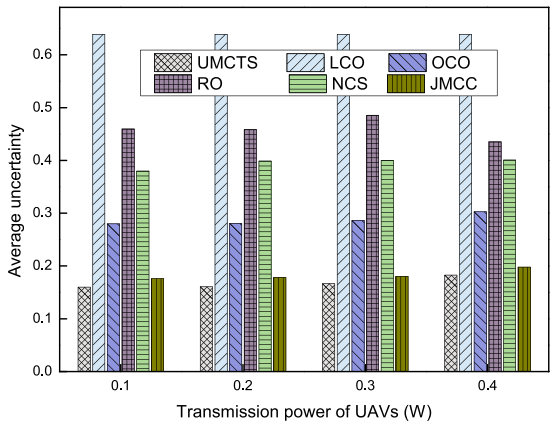


Fig. 9. Search performance under different number of cells.



(a)



(b)

Fig. 10. Average uncertainty under different processing capability and transmission power of UAVs. (a) Average uncertainty vs. processing capability of UAVs. (b) Average uncertainty vs. transmission power of UAVs.

uncertainty is already less than 0.1, which is a very small value, and no big performance gain will be achieved by increasing the number of UAVs. This interesting finding indicates that once a good average uncertainty performance requirement is met, the increase of the number of UAVs would not lead to a significant performance gain but a waste of resources. To reflect the effect of the number of cell on the search performance, we vary the search area to 300 m $\times$ 300 m, 400 m $\times$ 400 m, 400 m $\times$ 500 m, 400 m $\times$ 600 m, and 500 m $\times$ 500 m, such that there are 15 $\times$ 15, 20 $\times$ 20, 20 $\times$ 25, 20 $\times$ 30, and 25 $\times$ 25 cells, respectively. Fig. 9 shows that the average uncertainty of UMCTS, LCO, RO, NCS, and JMCC decreases when the number of cells increases. The underlying reason is that there are more undetected search areas when the number of cells increases. According to the definition of average uncertainty and the fact that the uncertainty of a cell will be reduced as the cell is searched repeatedly in formula (17), there would be more cells that are not searched or adequately searched when the number of cells increases, resulting in the increase of average uncertainty. More importantly, our proposed UMCTS outperforms LCO, OCO, RO, NCS, and JMCC in reducing the average uncertainty under different number of cells.

7) Effect of Processing Capability and Transmission Power of UAVs on Search Performance: Let the processing capability of UAVs vary from 1 $\times$ 10^9 cycles/s to 2.5 $\times$ 10^9 cycles/s, and the transmission power of UAVs vary from 0.1 W (i.e., 20 dBm)

to 0.4 W (i.e., 26.02 dBm), Fig. 10(a) shows that the average uncertainty of UMCTS, LCO, RO, NCS, JMCC increases with the increasing processing capability while that of OCO keeps unchanged. This is because more local computing energy will be consumed when f_n^l increases based on formula (7). Fig. 10(b) shows that the average uncertainty of UMCTS, OCO, RO, and NCS increases slightly with the increasing transmission power while that of LCO keeps unchanged. From both the two figures in Fig. 10, it is noteworthy that our proposed UMCTS outperforms other schemes under different processing capabilities and transmission power of UAVs.

VI. Conclusion and Future Work

This paper investigates the edge computing-enabled multi-UAV cooperative target search problem. Specifically, we first establish an edge computing enabled multi-UAV cooperative target search framework, where the mobility model of UAV and the search task computing and offloading models are established, and an uncertainty optimization problem is formulated. Considering round-trip energy consumption, offloading decision-making, and trajectory planning may contribute to the reduction in uncertainty, we propose an uncertainty minimization-based cooperative target search (UMCTS), where return energy consumption, offloading decision-making, and trajectory planning are optimized. Finally, we validate the performance of UMCTS by simulations. Simulation results show that UMCTS could achieve at least 89% performance gain on average uncertainty compared with the existing strategies. Also, the effects of different parameters on the search performance are further explored by simulations, which could guide the deployment and configuration of UAVs for practical search tasks.

In the future, we will consider a more general scenario where UAVs may move out of the communication range of GBSs. The cooperative offloading process would be investigated by cooperative network architectures of UAVs.

References

[1] L. Sun, L. Wan, and X. Wang, "Learning-based resource allocation strategy for industrial IoT in UAV-enabled MEC systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 5051–5040, Jul. 2021.

[2] B. Fei, W. Bao, X. Zhu, D. Liu, T. Men, and Z. Xiao, "Autonomous cooperative search model for multi-UAV with limited communication network," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 19346–19361, Oct. 2022.

[3] D. Ebrahimi, S. Sharafeddine, P.-H. Ho, and C. Assi, "Autonomous UAV trajectory for localizing ground objects: A reinforcement learning approach," *IEEE Trans. Mobile Comput.*, vol. 20, no. 4, pp. 1312–1324, Apr. 2021.

[4] X. Gu, D. Tang, and X. Huang, "Deep learning-based defect detection and recognition of a power grid inspection image," *Power System Protection Control*, vol. 49, no. 5, pp. 91–97, 2021.

[5] Z. Zhou, C. Zhang, C. Xu, F. Xiong, Y. Zhang, and T. Umer, "Energy-efficient industrial internet of UAVs for power line inspection in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2705–2714, Jun. 2018.

[6] A. C. Symington, S. Waharte, S. J. Julier, and N. Trigoni, "Probabilistic target detection by camera-equipped UAVs," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 4076–4081.

[7] T. Bai, J. Wang, Y. Ren, and L. Hanzo, "Energy-efficient computation offloading for secure UAV-edge-computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6074–6087, Jun. 2019.

[8] S. Garg, A. Singh, S. Batra, N. Kumar, and L. T. Yang, "UAV-empowered edge computing environment for cyber-threat detection in smart vehicles," *IEEE Netw.*, vol. 32, no. 3, pp. 42–51, May-Jun. 2018.

[9] S. Zhu et al., "Learning-based computation offloading approaches in UAVs-assisted edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 928–944, Jan. 2021.

[10] N. Zhao, Z. Ye, Y. Pei, Y.-C. Liang, and D. Niyato, "Multi-agent deep reinforcement learning for task offloading in UAV-assisted mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 6949–6960, Sep. 2022.

[11] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1927–1941, Sep. 2018.

[12] H. Guo and J. Liu, "UAV-enhanced intelligent offloading for Internet of Things at the edge," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2737–2746, Apr. 2020.

[13] J. Zhanget al., "Computation-efficient offloading and trajectory scheduling for multi-UAV assisted mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2114–2125, Feb. 2020.

[14] Q. Zhang, J. Chen, L. Ji, Z. Feng, Z. Han, and Z. Chen, "Response delay optimization in mobile edge computing enabled UAV swarm," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3280–3295, Mar. 2020.

[15] A. Khan, B. Rinner, and A. Cavallaro, "Cooperative robots to observe moving targets," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 187–198, Jan. 2018.

[16] J. Ni, L. Yang, P. Shi, and C. Luo, "An improved DSA-based approach for multi-AUV cooperative search," *Comput. Intell. Neurosci.*, vol. 2018, 2018, Art. no. 2186574.

[17] Y. Yang, A. A. Minai, and M. M. Polycarpou, "Decentralized cooperative search by networked UAVs in an uncertain environment," in *Proc. IEEE Amer. Control Conf.*, 2004, vol. 6, pp. 5558–5563.

[18] Y. Yang, M. M. Polycarpou, and A. A. Minai, "Multi-UAV cooperative search using an opportunistic learning method," *J. Dyn. Syst., Meas., Control*, vol. 129, pp. 716–728, 2007.

[19] G. B. Lamont, J. N. Slear, and K. Melendez, "UAV swarm mission planning and routing using multi-objective evolutionary algorithms," in *Proc. IEEE Symp. Comput. Intell. Multi-Criteria Decis.- Mak.*, 2007, pp. 10–20.

[20] J. Gu, T. Su, Q. Wang, X. Du, and M. Guizani, "Multiple moving targets surveillance based on a cooperative network for multi-UAV," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 82–89, Apr. 2018.

[21] J. R. Riehl, G. E. Collins, and J. P. Hespanha, "Cooperative search by UAV teams: A model predictive approach using dynamic graphs," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 4, pp. 2637–2656, Oct. 2011.

[22] P. Li and H. Duan, "A potential game approach to multiple UAV cooperative search and surveillance," *Aerosp. Sci. Technol.*, vol. 68, pp. 403–415, 2017.

[23] Z. Zhen, D. Xing, and C. Gao, "Cooperative search-attack mission planning for multi-UAV based on intelligent self-organized algorithm," *Aerosp. Sci. Technol.*, vol. 76, pp. 402–411, 2018.

[24] D. Luo, J. Shao, Y. Xu, Y. You, and H. Duan, "Coevolution pigeon-inspired optimization with cooperation-competition mechanism for multi-UAV cooperative region search," *Appl. Sci.*, vol. 9, no. 5, 2019, Art. no. 827.

[25] Z. Zhou, D. Luo, J. Shao, Y. Xu, and Y. You, "Immune genetic algorithm based multi-UAV cooperative target search with event-triggered mechanism," *Phys. Commun.*, vol. 41, 2020, Art. no. 101103.

[26] H. Duan, J. Zhao, Y. Deng, Y. Shi, and X. Ding, "Dynamic discrete pigeon-inspired optimization for multi-UAV cooperative search-attack mission planning," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 1, pp. 706–720, Feb. 2021.

[27] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, "Energy efficient resource allocation in UAV-enabled mobile edge computing networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4576–4589, Sep. 2019.

[28] X. Huang, X. Yang, Q. Chen, and J. Zhang, "Task offloading optimization for UAV-assisted fog-enabled Internet of Things networks," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1082–1094, Jan. 2022.

[29] A. Sacco, F. Esposito, G. Marchetto, and P. Montuschi, "Sustainable task offloading in UAV networks via multi-agent reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 5, pp. 5003–5015, May 2021.

[30] W. Chen, B. Liu, H. Huang, S. Guo, and Z. Zheng, "When UAV swarm meets edge-cloud computing: The QoS perspective," *IEEE Netw.*, vol. 33, no. 2, pp. 36–43, Mar./Apr. 2019.

[31] X. Cao, J. Xu, and R. Zhang, "Mobile edge computing for cellular-connected UAV: Computation offloading and trajectory optimization," in *Proc. 19th IEEE Int. Workshop Signal Process. Adv. Wireless Commun.*, 2018, pp. 1–5.

[32] J. Liuet al., "Minimization of offloading delay for two-tier UAV with mobile edge computing," in *Proc. 15th IEEE Int. Wireless Commun. Mobile Comput. Conf.*, 2019, pp. 1534–1538.

[33] M. Dai, Z. Su, Q. Xu, and N. Zhang, "Vehicle assisted computing offloading for unmanned aerial vehicles in smart city," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1932–1944, Mar. 2021.
[34] D. Callegaro and M. Levorato, "Optimal edge computing for infrastructure-assisted UAV systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1782–1792, Feb. 2021.
[35] A. Filippone, *Flight Performance of Fixed and Rotary Wing Aircraft*. Amsterdam, Netherlands: Elsevier, 2006.
[36] Q. Luo, C. Li, T. H. Luan, W. Shi, and W. Wu, "Self-learning based computation offloading for internet of vehicles: Model and algorithm," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 5913–5925, Sep. 2021.
[37] "DJI Consumer drones comparison." Accessed: Jan. 10 2022. [Online]. Available: https://www.dji.com/products/compare-consumer-drones
[38] E. W. Dijkstra et al., "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
[39] J. Chenet al., "Joint task assignment and spectrum allocation in heterogeneous UAV communication networks: A coalition formation game-theoretic approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 440–452, Jan. 2021.
[40] "Qualcomm, Paving the path to 5G: Optimizing commercial LTE networks for drone communication." Accessed: Sep. 27, 2022. [Online]. Available: https://www.qualcomm.com/news/onq/2016/09/paving-path-5g-optimizing-commercial-lte-networks-drone-communication
[41] 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, Mar. 2018.
[42] Y. Jack et al., "High performance chinese license plate recognition framework." Accessed: Jan. 23 2022. [Online]. Available: https://github.com/szad670401/HyperLPR
[43] Z. Xu, W. Yang, A. Meng, N. Lu, and H. Huang, "Towards end-to-end license plate detection and recognition: A large dataset and baseline," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 255–271.
[44] T. B. da Silva, R.P. dos Santos Chaib, C. S. Arismar, R. da Rosa Righi, and A. M. Alberti, "Toward future Internet of Things experimentation and evaluation," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8469–8484, Jun. 2022.
[45] J. Zhou, D. Tian, Z. Sheng, X. Duan, and X. Shen, "Joint mobility, communication and computation optimization for UAVs in air-ground cooperative networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 3, pp. 2493–2507, Mar. 2021.

Quyuan Luo received the Ph.D. degree in communication and information system from Xidian University, Xi'an, China, in 2020. He had been a visiting scholar with computer science, Wayne State University, USA from 2019 to 2020. He is currently an Assistant Professor with the Provincial Key Laboratory of Information Coding and Transmission, and with the School of Information Science and Technology, Southwest Jiaotong University, China. His current research interests include intelligent transportation systems, content distribution, edge computing, and resource allocation in vehicular networks. He is a recipient of the Excellent Doctoral Dissertation Award of China Education Society of Electronics (2021).

Tom H. Luan (Senior Member, IEEE) received the B.Eng. degree from Xi'an Jiaotong University, Xi'an, China, in 2004, the M.Phil. degree from The Hong Kong University of Science and Technology, Hong Kong, in 2007, and the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 2012. He is currently a Professor with the School of Cyber Engineering, Xidian University, Xi'an. Dr. Luan's research mainly focuses on the content distribution and media streaming in the vehicular ad hoc networks and peer-to-peer networking, protocol design and performance evaluation of wireless cloud computing and fog computing. Dr. Luan has authored/coauthored around 60 journal papers and 30 technical papers in conference proceedings, and awarded one US patent.

Weisong Shi (Fellow, IEEE) received the B.S. degree in computer engineering from Xidian University, Xi'an, China, in 1995, and the Ph.D. degree in computer engineering from the Chinese Academy of Sciences, in 2000. He is a Professor and Chair of the Department of Computer and Information Sciences at the University of Delaware (UD), where he leads the Connected and Autonomous Research (CAR) Laboratory. Dr. Shi is an internationally renowned expert in edge computing, autonomous driving and connected health. His pioneer paper entitled "Edge Computing: Vision and Challenges" has been cited more than 5000 times. Before he joins UD, he was a Professor at Wayne State University (2002-2022). He has published more than 270 articles in peer-reviewed journals and conferences and served in editorial roles for more than 10 academic journals and publications, including EIC of Smart Health, AEIC of IEEE Internet Computing Magazine. He is a fellow of IEEE, and a distinguished member of ACM.

Pingzhi Fan (Fellow, IEEE) received his MSc degree in computer science from the Southwest Jiaotong University, China, in 1987, and PhD degree in Electronic Engineering from the Hull University, U.K., in 1994. He is currently a distinguished Professor and Director of the institute of mobile communications, Southwest Jiaotong University, China, and a visiting Professor of Leeds University, U.K. (1997-), a guest Professor of Shanghai Jiaotong University (1999-). He is a recipient of the U.K. ORS Award (1992), the NSFC Outstanding Young Scientist Award (1998), IEEE VTS Jack Neubauer Memorial Award (2018), and 2018 IEEE SPS Signal Processing Letters Best Paper Award. His current research interests include vehicular communications, massive multiple access and coding techniques, etc. He served as general chair or TPC chair of a number of international conferences including VTC2016Spring, IWSDA2019, ITW2018 etc. He is the founding chair of IEEE Chengdu (CD) Section, IEEE VTS BJ Chapter and IEEE ComSoc CD Chapter. He also served as an EXCOM member of IEEE Region 10, IET(IEE) Council and IET Asia Pacific Region. He has published over 300 international journal papers and 8 books (incl. edited), and is the inventor of 25 granted patents. He is an IEEE VTS Distinguished Lecturer (2015-2019), and a fellow of IEEE, IET, CIE and CIC.