Intelligent Computation Offloading and Trajectory Planning for 3D Target Search in Low-Altitude Economy Scenarios

Helin Yang, *Senior Member, IEEE*, Mengting Zheng, *Student Member, IEEE*, Ziling Shao, *Student Member, IEEE*, Yifu Jiang, and Zehui Xiong, *Senior Member, IEEE*

***Abstract*—The low-altitude economy provides new opportunities for target search, as unmanned aerial vehicles (UAVs) improve search efficiency through aerial surveillance and data collection. UAVs support overhead views of the search area, while ground vehicles (GVs) offer side views, facilitating effective collaboration in wireless networks. However, the limited battery life restrict UAVs from performing computation-intensive and latency-sensitive tasks, and their wireless communication links are susceptible to jamming attacks. Therefore, this paper proposes a joint resource scheduling approach for an edge computing enabled multi-UAV multi-GV cooperative target search system under intelligent jamming attacks. Specifically, the approach aims to minimize the uncertainty of the search area by jointly optimizing ground base station (GBS) association, task offloading rate, and 3D trajectory control. Since the problem is non-convex and the intelligent jamming behavior is dynamic, a multi-agent softmax deep double deterministic policy gradients (MA-SD3) approach is proposed to effectively perform resource management for target search and resist jamming attacks. Simulation results demonstrate that the proposed approach outperforms the baseline approaches under different settings.**

***Index Terms*—Unmanned aerial vehicle, ground vehicle, target search, edge computing, anti-jamming, reinforcement learning.**

I. Introduction

In recent years, target search has been a hot topic, with widespread applications in various fields, such as disaster management, search and rescue operations, and environmental monitoring. Low-altitude economy refers to an economic form that includes various low-altitude flight activities of unmanned and manned aircraft, generating significant economic and social value. It presents new opportunities for target search, where unmanned aerial vehicles (UAVs) enhance search efficiency through aerial surveillance and data collection [1], [2]. However, UAVs provide overhead views in target search but lack side perspectives, which can be addressed by integrating ground vehicles (GVs) to offer side views. Wireless communication enables multi-GV and multi-UAV to exchange real-time data for collaborative target search [3], which has been emerged as a promising paradigm to effectively support air-to-ground communication networks.

A. Literature Review

In UAV-assisted communication systems, the short battery life and insufficient computational power restrict UAVs from handling computation-intensive and latency-sensitive search tasks [4]. To tackle these challenges, edge computing (EC) has emerged as a key technology, offloading data to edge servers for computation, which alleviates data growth pressure and enhances computational efficiency [5], [6].

The edge computing enabled multi-UAV multi-GV can improve target detection, surveillance, and anomaly identification to better meet the complex demands of search missions in dynamic environments by strategically allocating communication resources. Recently, significant efforts have focused on resource optimization for edge computing in UAV and GV networks. In the multi-UAV-aided communication scenario, the authors of [7] aimed to minimize system latency and energy consumption by optimizing UAV positions, association, and transmit power. In the vehicle edge computing network, the authors of [8] introduced an efficient offloading scheme to maximize total utility. In both studies, UAVs and GVs act as edge servers. However, both UAVs and GVs generate data by capturing images in target search scenario, which can be processed locally or offloaded to edge servers. Several studies have explored edge computing enabled UAVs, showing that offloading tasks to ground edge servers alleviates the computational burden on UAVs and enhances their processing capabilities [9]–[11]. The research on integrating UAVs, GVs, and edge computing for target search remains limited.

The 3D target search in low-altitude economy scenario offloads computational tasks via a wireless transmission link, which is particularly vulnerable to jamming attacks. Reinforcement Learning (RL) has been widely employed in anti-jamming due to its adaptability to dynamic environments. For instance, in [12] and [13], the authors used an RL-based approach to solve UAV anti-jamming challenges. In the case of UAV-assisted edge computing, the authors of [14] applied deep reinforcement learning (DRL) to optimize UAV hover point selection and reduce the impact of jamming attacks.

This work was supported by the National Natural Science Foundation of China under Grants No. 62371408, Fujian Provincial Natural Science Foundation of China under Grant No. 2024J09002, Xiaomi Young Talents Program, the National Research Foundation, Singapore and Infocomm Media Development Authority under its Future Communications Research & Development Programme. The research is also supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (MOE-T2EP20221-0017). (Corresponding author: *Yifu Jiang*)

H. Yang, M. Zheng, and Z. Shao are with the Department of Information and Communication Engineering, Xiamen University, Xiamen, China 361005 (e-mail: helinyang066@xmu.edu.cn, {mtingzheng, shaoziling}@stu.xmu.edu.cn,).

Y. Jiang is with the Department of Economics Analysis, University of Zaragoza, Zaragoza 50001, Spain (e-mail: yifu.jiang.2016@mm.smu.edu.sg).

Z. Xiong is with the Pillar of Information Systems Technology and Design, Singapore University of Technology and Design, Singapore 487372 (email: zehui_xiong@sutd.edu.sg).

Additionally, the authors of [15] employed a multi-agent DRL (MA-DRL) method for channel selection to avoid jamming channel. These studies highlight the potential of DRL to effectively resist malicious jamming attacks.

B. Main Contribution

Motivated by the above discussions, we design an edge computing enabled multi-UAV multi-GV cooperative target search model under intelligent jamming attacks. Specifically, we propose a joint resource scheduling approach to optimize ground base station (GBS) association, 3D trajectory control, and task offloading rate to minimize the uncertainty of search area. Due to the non-convex optimization problem and dynamic jamming behavior, we convert it into a multi-agent problem by treating each UAV-GV pair as an agent and develop a multi-agent softmax deep double deterministic policy gradients (MA-SD3) approach to perform resource management. Simulation results show that the proposed approach outperforms the benchmark approaches under different real-world settings.

II. System Model and Problem Formulation

A. System Model

As shown in Fig. 1, we consider 3D target search in low-altitude economy scenarios under intelligent jamming attacks within wireless communication systems. The search area Ω is a rectangular plane divided into $L_x \times L_y$ discrete cells. Each cell is identified by its center denoted as $O_{x,y}$, where $\big(x \in \{1, 2, \ldots, L_x\}, y \in \{1, 2, \ldots, L_y\}\big)$. The systems search time horizon T is divided into N time slots of varying durations, denoted by the set $\mathcal{N} = \{1, 2, ..., N\}$, where $t[n]$ is the duration of time slot n.

We regard a UAV and a GV as a pair to search for a cell at the same time, and the set of such pairs is denoted as $\mathcal{P} = \{p_1^i, \ldots, p_k^i, \ldots, p_K^i\}, i \in \{U, V\}$, where K is the number of pairs. When $i = U$, it indicates UAV p_k^U in pair p_k^i, which flies over cells at an altitude of $H_{p_k^U}[n]$ and captures top-view images of a cell. When $i = V$, it means GV p_k^V in pair p_k^i, which captures side-view images of the cell simultaneously, with an altitude of $H_{p_k^V}[n] = 0$. The UAV and GV in each pair have the same horizontal trajectory, and the position of pair p_k^i at time slot n is given by $\mathbf{u}_{p_k^i}[n] = (x_{p_k^i}[n], y_{p_k^i}[n], H_{p_k^i}[n])$. The set of horizontal movement directions $\mathcal{C}_{p_k^i}[n]$ for UAV-GV pair p_k^i at time slot n includes eight directions: north, northeast, east, southeast, south, southwest, west, and northwest. However, the available directions are constrained by their position, as movement range cannot exceed the boundaries of the search area. Both the UAVs and GVs as users can offload image processing tasks to GBS equipped with edge servers, which provides powerful computing capacities. The set of GBSs is denoted as $\mathcal{Q} = \{1, \ldots, q_m, \ldots, q_M\}$, where M is the number of GBSs. The position of GBS q_m is denoted as $\mathbf{v}_{q_m} = (x_{q_m}, y_{q_m}, H_b)$. Let $D_{O_{x,y}} = \{D^U_{O_{x,y}}, D^V_{O_{x,y}}\}$ represent the search task for cell $O_{x,y}$, where $D^U_{O_{x,y}}$ and $D^V_{O_{x,y}}$ are the search task sizes for UAVs top-view and GVs side-view perspective, respectively.

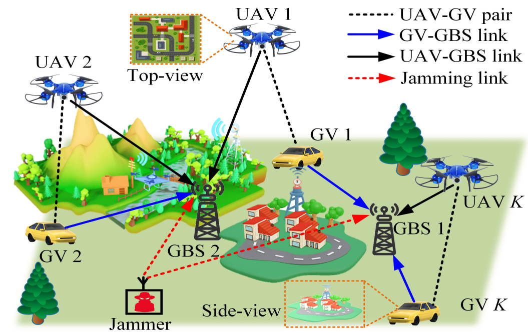

Fig. 1. The wireless communication system model of edge computing enabled multi-UAV multi-GV cooperative target search under jamming.

B. The Intelligent Jamming Model

We consider the ground jammer (GJ) as an intelligent agent, which continuously observes the positions and computation offloading status of UAVs and GVs. Based on this information and employing a greedy decision-making strategy, the GJ adjusts its movement direction and distance to maximize the transmission delay. Its position at time slot n is given by $\mathbf{w}_{GJ}[n] = (x_{GJ}[n], y_{GJ}[n], 0)$. This process can be modeled as a game between UAV-GV pairs and the jammer.

C. Communication Model

Since the UAV operates at a relatively high altitude, it is likely to transmit line-of-sight (LoS) signals to the GBS. According to the probabilistic path loss model, the LoS probability between UAV p_k^U and GBS q_m is given by $\mathrm{Pr}^{\mathrm{LoS}}_{p_k^U,q_m}[n] = (1 + a_1 \exp(-a_2(\theta_{p_k^U,q_m}[n] - a_1)))^{-1}$, where a_1 and a_2 are constants that depend on the environment conditions, and $\theta_{p_k^U,q_m}[n]$ is the elevation angle. Consequently, the channel gain between UAV p_k^U and GBS q_m is written as

$$h_{p_k^U,q_m}[n] = 10^{-\frac{(\eta_{\mathrm{LoS}}-\eta_{\mathrm{NLoS}})\mathrm{Pr}^{\mathrm{LoS}}_{p_k^U,q_m}[n]+\eta_{\mathrm{NLoS}}}{10}} \left(\frac{4\pi f d_{p_k^U,q_m}[n]}{c}\right)^{-2}, \quad (1)$$

where $d_{p_k^U,q_m}[n]$ is the distance from UAV p_k^U to GBS q_m, η_{LoS} and η_{NLoS} represent the propagation losses for LoS and NLoS links, respectively.

Similar to [7], the channels from GV to GBS and from GJ to GBS are characterized as independent Rayleigh fading. Additionally, α^V and α^J are the path-loss exponents of GV-to-GBS and GJ-to-GBS channels, respectively. $g_{p_k^V,q_m}$ and $g^J_{q_m}$ represent the corresponding Rayleigh fading coefficient, while β_0 denotes the reference channel gain at distance 1m. Thus, the channel gain between GV p_k^V and GBS q_m is written as $h_{p_k^V,q_m}[n] = \beta_0 g_{p_k^V,q_m}(\|\mathbf{v}_{q_m} - \mathbf{u}_{p_k^V}[n]\|)^{-\alpha^V}$, and the channel gain between GJ and GBS q_m is given by $h^J_{q_m}[n] = \beta_0 g^J_{q_m}(\|\mathbf{v}_{q_m} - \mathbf{w}_{GJ}[n]\|)^{-\alpha^J}$. Therefore, the transmission rate between pair p_k^i and GBS q_m at time slot n

can be defined as

$$R_{p_k^i,q_m}[n] = \frac{W}{\varepsilon_{q_m}[n]} \log_2 \left(1 + \frac{P_{p_k^i} h_{p_k^i,q_m}[n]}{\sigma^2 + P_J h_{q_m}^J[n]}\right), i \in \{U,V\}, \quad (2)$$

where σ^2 gives the noise power, W is the total bandwidth of each GBS, $\varepsilon_{q_m}[n]$ represents the number of connections to GBS q_m at time slot n, $P_{p_k^i}$ and P_J are the transmission power of pair p_k^i and GJ, respectively.

D. Computation Model

The system uses a partial offloading mode, with the offloading rate of UAV p_k^U and GV p_k^V to the associated GBS q_m given by $\lambda_{p_k^U,q_m}[n]$ and $\lambda_{p_k^V,q_m}[n]$, respectively. Therefore, the local computing delay required by pair p_k^i is denoted as

$$T_{p_k^i}^{\mathrm{Loc}}[n] = \frac{(1-\lambda_{p_k^i,q_m}[n]) D_{O_{x,y}}^i C}{f_{p_k^i}}, i \in \{U,V\}, \quad (3)$$

where $f_{p_k^i}$ is the computation capability of pair p_k^i, and C represents the number of CPU cycles required to process one bit. The association indicator variable $b_{p_k^U,q_m}[n]$ and $b_{p_k^V,q_m}[n]$ indicate whether UAV p_k^U and GV p_k^V are associated with GBS q_m or not at time slot n. Hence, the transmission delay from p_k^i to the associated q_m is given by

$$T_{p_k^i,q_m}^{\mathrm{Tr}}[n] = b_{p_k^i,q_m}[n] \frac{\lambda_{p_k^i,q_m}[n] D_{O_{x,y}}^i}{R_{p_k^i,q_m}[n]}, i \in \{U,V\}. \quad (4)$$

The UAV and GV in the pair follow the same horizontal trajectory with movement time $t_{p_k}^{move}[n]$. The duration of time slot n for pair p_k is calculated as $t_{p_k}[n] = t_{p_k}^{move}[n] + \max(T_{p_k^U}^{\mathrm{Loc}}[n], T_{p_k^V}^{\mathrm{Loc}}[n], T_{p_k^U,q_m}^{\mathrm{Tr}}[n], T_{p_k^V,q_m}^{\mathrm{Tr}}[n])$. Thus, the duration of time slot n is $t[n] = \max(t_{p_1}[n], t_{p_2}[n], \ldots, t_{p_K}[n])$. The energy consumption of the UAV p_k^U at time slot n is denoted as $E_k[n]$.

E. Problem Formulation

Each cell has uncertainty $u(O_{x,y}, n) \in [0,1]$ which reflects the reliability of target detection results. We denote the accuracy of target detection as ρ, and $u(O_{x,y}, n)$ will decrease as cell $O_{x,y}$ is searched at time slot n, expressed as [16]

$$u(O_{x,y}, n+1) = (1-\rho) u(O_{x,y}, n), \quad (5)$$

Our objective is to minimize the uncertainty of the search area by jointly optimizing computation offloading (GBS association $\mathbf{b}$ and task offloading rate $\boldsymbol{\lambda}$) and 3D trajectory planning (horizontal movement direction of UAV-GV pairs $\mathbf{c}$ and the altitude of UAVs $\mathbf{H}$). The optimization problem is formulated as follows

$$\begin{aligned}
\textbf{(P1)}: & \underset{\{\mathbf{b},\mathbf{c},\mathbf{H},\boldsymbol{\lambda}\}}{\text{minimize}} \sum\nolimits_{O_{x,y}\in\Omega} u(O_{x,y}, N) \\
\text{s.t. } & (\mathrm{C1}): \ \sum\nolimits_{n=1}^{N} t[n] = T, \\
& (\mathrm{C2}): \ c_{p_k^i}[n] \in \mathcal{C}_{p_k^i}[n], \forall k, n, i \in \{U,V\}, \\
& (\mathrm{C3}): \ \sum\nolimits_{m=1}^{M} b_{p_k^i,q_m}[n] = 1, \forall k, n, i \in \{U,V\}, \\
& (\mathrm{C4}): \ |H_{p_k^U}[n+1] - H_{p_k^U}[n]| \le t_{p_k}^{move}[n] V_{\max}^{\mathrm{z}}, \forall k, n, \\
& (\mathrm{C5}): \ H_{\min} \le H_{p_k^U}[n] \le H_{\max}, \forall k, n.
\end{aligned} \quad (6)$$

In (**P1**), constraint (C1) defines the total time allocated for systems target search. Constraint (C2) ensures that the UAVs and GVs select movement directions within boundary constraints. Moreover, constraint (C3) indicates that both the UAVs and GVs can offload computation to only one GBS servers per time slot. Constraints (C4) and (C5) limit the maximum vertical change in altitude for each time slot and define the allowable altitude range for UAVs.

III. MA-SD3-Based Computation Offloading and Trajectory Planning Approach

The problem formulated in (6) is a mixed-integer nonlinear programming problem in highly dynamic environment. Therefore, an MA-DRL approach is proposed to capture interactions and dependencies among agents to optimize the solution.

A. Multi-Agent Markov Decision Process

In this study, each UAV-GV pair is treated as an agent that collaborates to cover the target search area as efficiently as possible. This problem can be modeled as a multi-agent Markov decision process (MDP), which consists of state space S, action space A, and reward function R.

State space: The state $s_k[n]$ represents the information currently observed by agent k at time slot n. This includes the uncertainty $u(O_{x,y}, n)$ of all cells, the remaining execution time $T_r[n]$, the positions $\mathbf{u}_{p_k^i}[n]$ of the UAV-GV pair p_k^i, and the total energy consumption $E_k^{total}[n] = \sum_{i=1}^{n} E_k[i]$ of the UAV p_k^U. Then the state is given by $s_k[n] = \{u(O_{1,1}, n), ..., u(O_{L_x,L_y}, n), T_r[n], \mathbf{u}_{p_k^i}[n], E_k^{total}[n]\}_{k\in\mathcal{K}}$.

Action space: Each agent selects the GBS associated with the UAVs and GVs $b_{p_k^i,q_m}[n]$, determines the offloading rate $\lambda_{p_k^i,q_m}[n]$ of the UAVs and GVs, controls the horizontal movement direction $c_{p_k^i}[n]$ of the UAV-GV pairs, and designs the altitude $H_{p_k^U}[n]$ of UAVs. Thus, the action space for agent k at time slot n is denoted as $a_k[n] = \{b_{p_k^i,q_m}[n], \lambda_{p_k^i,q_m}[n], c_{p_k^i}[n], H_{p_k^U}[n]\}_{k\in\mathcal{K}}$.

Reward function: We incorporate both uncertainty reduction revenue and delay cost under energy constraints into the reward design. The UAV battery capacity is set to E_b. When $E_k^{total}[n] < E_b$, the UAV collaborates with the GV to conduct target search, and the reward for uncertainty reduction is defined as $\rho u(O_{x,y}, n)$. Otherwise, only the GV conducts target search, and the reward for uncertainty reduction is $\rho u(O_{x,y}, n)/2$. The reward function is defined as where λ_1 is weighing factor.

$$r_k[n] = \begin{cases} \rho u(O_{x,y}, n) - \lambda_1 t[n], E_k^{total}[n] < E_b, \\ \frac{\rho}{2} u(O_{x,y}, n) - \lambda_1 t[n], E_k^{total}[n] \ge E_b, \end{cases} \quad (7)$$

B. MA-SD3-Based Resource Scheduling Solution

We address the problem (**P1**) using the MA-SD3 approach, a DRL method designed for multi-agent environments. The MA-SD3 employs a centralized training with decentralized execution method. Each agent is aware of the strategies and states of all other agents during training, while it relies solely on its local observations to make decisions during execution. Each agent in the MA-SD3 employs the SD3 algorithm, which builds on the deep deterministic policy gradient (DDPG) and twin delayed deep deterministic policy gradient (TD3) algorithms. Compared to DDPG, SD3 benefits from TD3 by using the minimum of two target networks for value estimation, reducing overestimation bias. Additionally, SD3 uses a softmax operator to refine the action-value function for continuous control, addressing the underestimation bias tendency of TD3. Furthermore, SD3 uses a two-actor network to overcome the inefficient exploration problem associated with TD3's single-actor network.

Agent k uses the SD3 algorithm for training, which involves two actor networks with weights $\{\theta_k^t\}_{t=1,2}$ for training and learning the model and two target actor networks with weights $\{\theta_k'^t\}_{t=1,2}$ for preventing tampering with the training data. It also includes two critic networks with weights $\{\delta_k^t\}_{t=1,2}$ and two target critic networks with weights $\{\delta_k'^t\}_{t=1,2}$. The actor networks π_k output an action distribution based on the current policy, and the critic networks Q_k evaluate the value of the actions chosen by the actor. By comparing the estimated Q-value $Q_k(s_k, \pi_k(s_k;\theta_k^t);\delta_k^t)$, the execution action will be selected from the larger Q-value $\max(Q_k(s_k, \pi_k(s_k;\theta_k^1);\delta_k^1), Q_k(s_k, \pi_k(s_k;\theta_k^2);\delta_k^2))$. Hence, this approach can enhances the exploration capability.

After executing the actions $\mathbf{a}$ and proceeding to the next state $\mathbf{s}'$, the experience $\{\mathbf{s}, \mathbf{s}', \mathbf{a}, \mathbf{r}\}$ will be stored in the replay buffer $\mathcal{B}$. Then, we sample a mini-batch size M_b of transitions $\{\mathbf{s}^z, \mathbf{s}'^z, \mathbf{a}^z, \mathbf{r}^z\}$, and the actor network is updated by policy gradient descent, denoted as

$$\nabla_{\theta_k^t} J(\theta_k^t) = \frac{1}{M_b} \sum_{z=1}^{M_b} [\nabla_{\theta_k^t} \pi_k(s_k^z;\theta_k^t) \nabla_{a_k} Q_k \\ (\mathbf{s}^z, a_1^z, \ldots, a_K^z; \delta_k^t) \Big|_{a_k^z = \pi_k(s_k^z;\theta_k^t)}], \tag{8}$$

where $Q_k(\mathbf{s}^z, a_1^z, \ldots, a_K^z; \delta_k^t)$ approximates the centralized action-value function, and $\pi_k(s_k^z;\theta_k^t)$ approximates the policy function. It uses the minimum of two target critic networks to estimate the target action-value function $Q_k'(\mathbf{s}'^z, \mathbf{a}'^z) = \min(Q_k'(\mathbf{s}'^z, \mathbf{a}'^z; \delta_k'^1), Q_k'(\mathbf{s}'^z, \mathbf{a}'^z; \delta_k'^2))$. The softmax operator is applied to reduce overestimation and underestimation bias. Hence, the target Q-value can be expressed as

$$y_k^z \leftarrow r_k^z + \gamma softmax_\beta \left(Q_k'(\mathbf{s}'^z, \cdot) \right), \tag{9}$$

where γ represents the discount factor. Consequently, the weights $\{\delta_k^t\}_{t=1,2}$ of the critic networks are trained by minimizing a loss function defined as

$$L_k(\delta_k^t) = \frac{1}{M_b} \sum_{z=1}^{M_b} \Big(y_k^z - Q_k(\mathbf{s}^z, \mathbf{a}^z; \delta_k^t) \Big)^2. \tag{10}$$

The MA-SD3 approach is summarized in **Algorithm 1**.

Algorithm 1 MA-SD3-Based Approach

Initialize: Each agents actor network π_k with weights $\{\theta_k^t\}_{t=1,2}$ and critic networks Q_k with weights $\{\delta_k^t\}_{t=1,2}$.
Initialize: Target networks parameters $\theta_k'^t \leftarrow \theta_k^t$, $\delta_k'^t \leftarrow \delta_k^t$.
1: **for each** *episode* **do**
2: Initialize the state $\mathbf{s}[1] = (s_1[1], s_2[1], \ldots, s_K[1])$.
3: **for** $n = 1$ to N_s **do**
4: Each agent selects actions $a_k[n] \sim \pi_k(s_k[n];\theta_k^t) + \varepsilon$.
5: Execute $\mathbf{a}[n] = (a_1[n], a_2[n], ..., a_K[n])$, obtain next state $\mathbf{s}[n+1]$, and get reward $\mathbf{r}[n] = (r_1[n], r_2[n], ..., r_K[n])$.
6: Store $\{\mathbf{s}[n], \mathbf{a}[n], \mathbf{r}[n], \mathbf{s}[n+1]\}$ in the replay buffer $\mathcal{B}$.
7: **for** each agent k **do**
8: Sample mini-batch $\{\mathbf{s}^z, \mathbf{s}'^z, \mathbf{a}^z, \mathbf{r}^z\}$ of size M_b of transitions from $\mathcal{B}$.
9: Update actor and critic networks by (8) and (10).
10: Update target networks by using soft update rule.
11: **End for**
12: **End for**
13: **End for**

C. Complexity Analysis

Each actor and critic network consists of an input layer, an output layer, and three hidden layers, with the actor network having l_i^a neurons and the critic network having l_i^c neurons in the i-th hidden layer. Let N_e, N_s, M_b, and D denote the number of training episodes, the number of steps in each episode, the size of the mini-batch, and the replay buffer size, respectively. The computational complexity for K agents in the MA-SD3 approach is given by $\mathcal{O}((N_e N_s - D) K M_b (\sum_{i=0}^{3} 4 l_i^a l_{i+1}^a + \sum_{i=0}^{3} 4 l_i^c l_{i+1}^c))$.

IV. Simulation Results and Analysis

In the simulation, the search area Ω is 200×200 m^2, divided into 10×10 cells. Four UAV-GV pairs are positioned at $[1,1]$, $[10,10]$, $[1,10]$, and $[10,1]$, with a horizontal speed of $V^{\mathrm{h}} = 15$ m/s and a maximum vertical speed of $V_{\max}^{\mathrm{z}} = 10$ m/s. Additionally, the UAVs fly at an altitude range of $[50, 100]$ m. Two GBS are located at $[5,1]$ and $[6,10]$. A GJ starts at the origin with a fixed speed of 10 m/s and transmits jamming signals with a power of $P_J = 0.15$ W. The total area search time is $T = 240$ s, and target detection accuracy is $\rho = 0.6$.

We evaluate the performance of the proposed MA-SD3-based computation offloading and trajectory planning approach (called MA-SD3-COTP) compared with four baseline schemes: 1) The SD3-based computation offloading and trajectory planning approach (called SD3-COTP), which treats all pairs as a single agent for training. 2) The MA-TD3-based computation offloading and trajectory planning approach (called MA-TD3-COTP). 3) The MA-DDPG-based computation offloading and trajectory planning approach (called MA-DDPG-COTP). 4) The MA-SD3 approach is used for trajectory planning optimization (called MA-SD3-TP).

The convergence performance of the proposed MA-SD3-COTP approach and benchmark approaches in terms of average uncertainty is shown in Fig. 2. The SD3-COTP approach is more susceptible to local optima in complex large-scale scenarios, and its single-agent perspective may lead to greater fluctuations. In contrast, MA-DRL approach improves system performance and adaptability through the collaboration and competition of multiple agents. The MA-SD3-COTP approach outperforms MA-TD3-COTP and MA-DDPG-COTP by leveraging SD3 for agent training, which uses a softmax policy to model action probabilities, enabling more flexible action

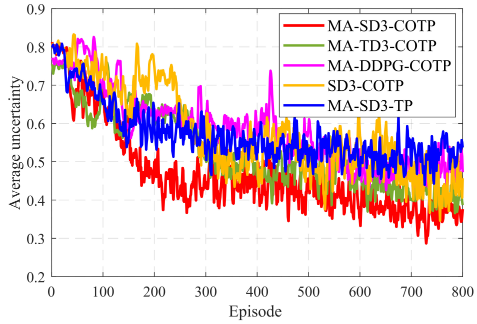

Fig. 2. Convergence performance comparisons.

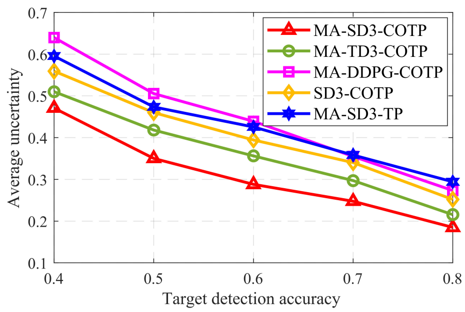

Fig. 3. Comparison of average uncertainty at varying detection accuracy.

exploration. Additionally, compared to MA-SD3-TP, the MA-SD3-COTP approach allows for more efficient resource allocation and effective resistance against jamming attacks.

Fig. 3 plots the average uncertainty of the search area versus target detection accuracy for five approaches with four UAV-GV pairs. As expected, the average uncertainty of all approaches decreases monotonically with increased target detection accuracy, though the rate of decrease becomes progressively slower. This is because higher target detection accuracy enables more precise identification of the search area, thereby reducing uncertainty. Moreover, it is evident that the MA-SD3-COTP approach consistently shows lower average uncertainty compared to MA-TD3-COTP, MA-DDPG-COTP, SD3-COTP and MA-SD3-TP, with reductions of up to 19%, 34%, 27%, and 32%, respectively, which demonstrates the effectiveness of the proposed approach.

In Fig. 4, we demonstrate the impact of the number of UAV-GV pairs on the average uncertainty of the search area with a target detection accuracy of 0.6. We observe that search performance improves when more UAV-GV pairs are involved in the search process. However, once the uncertainty is sufficiently reduced, excess UAVs and GVs provide minimal performance gains and lead to inefficient resource utilization. Furthermore, compared with other approaches, our proposed approach allows UAV-GV pairs to collaborate efficiently, resulting in a more significant reduction in search area uncertainty.

V. Conclusion

In this paper, we explore the resource scheduling for an edge computing enabled multi-UAV multi-GV cooperative target search system under intelligent jamming attacks. Specifically, we propose the MA-SD3 approach to minimize the uncertainty

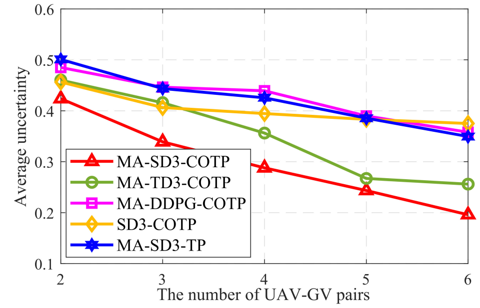

Fig. 4. Comparison of average uncertainty at varying pair numbers.

of the search area by jointly optimizing GBS association, 3D trajectory control, and task offloading rate. Simulation results verify the effectiveness of the proposed approach.

References

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

[2] B. Zheng and F. Liu, "Random signal design for joint communication and SAR imaging towards low-altitude economy," *IEEE Wireless Commun. Lett.*, vol. 13, no. 10, pp. 2662-2666, Oct. 2024.

[3] J. Li, G. Deng, C. Luo, Q. Lin, Q. Yan, and Z. Ming, "A hybrid path planning method in unmanned air/ground vehicle (UAV/UGV) cooperative systems," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9585-9596, Dec. 2016.

[4] Z. Kuang, Y. Pan, F. Yang, and Y. Zhang, "Joint task offloading scheduling and resource allocation in airCground cooperation UAV-enabled mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 73, no. 4, pp. 5796-5807, Apr. 2024.

[5] 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, Sept. 2018.

[6] D. Wang, J. Tian, H. Zhang, and D. Wu, "Task offloading and trajectory scheduling for UAV-enabled MEC networks: An optimal transport theory perspective," *IEEE Wireless Commun. Lett.*, vol. 11, no. 1, pp. 150-154, Jan. 2022.

[7] J. Chen et al., "Deep reinforcement learning based resource allocation in multi-UAV-aided MEC networks," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 296-309, Jan. 2023.

[8] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11158-11168, Nov. 2019.

[9] Q. Luo, T. H. Luan, W. Shi, and P. Fan, "Deep reinforcement learning based computation offloading and trajectory planning for multi-UAV cooperative target search," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 2, pp. 504-520, Feb. 2023.

[10] 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.

[11] 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.

[12] L. Xiao, Y. Ding, J. Huang, S. Liu, Y. Tang, and H. Dai, "UAV anti-jamming video transmissions with QoE guarantee: A reinforcement learning-based approach," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 5933-5947, Sept. 2021.

[13] Z. Lv, L. Xiao, Y. Du, G. Niu, C. Xing, and W. Xu, "Multi-agent reinforcement learning based UAV swarm communications against jamming," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 9063-9075, Dec. 2023.

[14] S. Liu, H. Yang, L. Xiao, M. Zheng, H. Lu, and Z. Xiong, "Learning-based resource management optimization for UAV-assisted MEC against jamming," *IEEE Trans. Commun.*, vol. 72, no. 8, pp. 4873-4886, Aug. 2024.

[15] Z. Shao, H. Yang, L. Xiao, W. Su, Y. Chen, and Z. Xiong, "Deep reinforcement learning-based resource management for UAV-assisted mobile edge computing against jamming," *IEEE Trans. Mob. Comput.*, early access, Jul. 29, 2024, doi: 10.1109/TMC.2024.3432491.

[16] 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, no. 5, pp. 716C728, Jan. 2007.