Cooperative Dual-Task Path Planning for Persistent Surveillance and Emergency Handling by Multiple Unmanned Ground Vehicles

JunQi Zhang, *Senior Member, IEEE*, YunZhe Wu, and MengChu Zhou, *Fellow, IEEE*

***Abstract*— Road network persistent surveillance requires a swarm of unmanned ground vehicles (UGVs) to repeatedly surveil a sequence of places called viewpoints on a road network and detect randomly occurring events and problems at viewpoints. In most existing work, UGVs are responsible for detection but not handling emergencies like extinguishing fires, capturing intruders, or managing pollution. Hence, existing methods fail to perform both persistent surveillance and emergency handling. This work copes with both and proposes a cooperative dual-task path planning method for multiple UGVs. UGVs first treat all the viewpoints as candidates and compute their rewards to select their targets. Emergencies have a higher reward, allowing UGVs to solve dual tasks and prioritize emergencies. A heuristic protocol is then employed to choose appropriate UGVs for emergencies. Local path planning is finally considered when UGVs plan paths to targets to avoid deadlock and backtracking. Thus, a necessary and appropriate subswarm of UGVs with the nearest distance to emergencies are selected to handle them, while others continue surveilling viewpoints for potential emergencies, avoiding any long delayed handling of emergencies. The performance of the proposed method is evaluated in three road networks. The simulation and analysis results demonstrate its superiority over the state-of-the-art surveillance methods.**

***Index Terms*— Dual-task, multi-robot systems, road network path planning.**

I. Introduction

COOPERATIVE surveillance with mobile robots intends to update environmental status and ensure safety, which is a fundamental mission of multi-robot systems and widely applied in epidemic prevention [1], territory guarding [2], disaster management [3], and environment monitoring [4], [5], [6], [7], [8]. A significant amount of work has been done to solve surveillance problems whose aims are to keep environmental security and acquire situational information. They are classified into persistent and adversarial ones [9].

A. Persistent Surveillance

Persistent surveillance requires robots to visit important positions called viewpoints as frequently as possible and is applied in crowd movement surveillance [10], precision agriculture management [11], environment monitoring [7] and data acquisition [12]. In this problem, there are events appearing at viewpoints randomly and unpredictably. However, their appearance time intervals are considered persistent and statistical [13]. Persistent surveillance aims to drive robots to detect events before they disappear by frequently surveilling the environment [14] and reducing the time interval between two consecutive visits for each viewpoint. Thus, the main evaluation criteria for such tasks are time and moving cost [15]. These methods share a common feature: mobile robots perform persistent cyclic visits to viewpoints with deterministic strategies. They assume that the environmental status changes over time [16]. Hence they work in a stable environment without considering emergency handling.

Existing persistent surveillance methods are divided into distributed and centralized ones. The former enables each robot to make its own independent decision, including reactive [17], learning-based [18], and auction-based methods [19]. Reactive methods use locally shared information to make decisions, such as ant colony [20] and greedy strategy [21]. Although these methods are computationally efficient, they can easily fall into local optima [22]. Learning-based ones use probabilistic models, such as Markov decision [18] and Bayesian learning [23]. They combine such models with reinforcement learning [24] and allow robots to adjust their strategies to dynamic environment changes. However, they require significant training cost [9]. In auction-based methods, robots negotiate with others to exchange viewpoints for surveillance [25]. The learning and auction methods are much more complex than reactive ones [26]. Centralized methods require a coordinator to assign viewpoints and paths to all robots. They can be classified into cognitive coordinated [26], cyclic [27], and partition-based [28] ones. In cognitive coordination ones, a central coordinator has global information and assigns the next target viewpoint to robots with the

Manuscript received 26 July 2023; revised 28 January 2024 and 13 May 2024; accepted 26 June 2024. Date of publication 22 July 2024; date of current version 1 November 2024. This work was supported by the Innovation Program of Shanghai Municipal Education Commission under Grant 2021010700007E00098. The Associate Editor for this article was S. A. Birrell. *(Corresponding author: MengChu Zhou.)*

JunQi Zhang and YunZhe Wu are with the Department of Computer Science and Technology, Key Laboratory of Embedded System and Service Computing, Ministry of Education, Shanghai Electronic Transactions and Information Service Collaborative Innovation Center, Tongji University, Shanghai 200092, China (e-mail: zhangjunqi@tongji.edu.cn; parks_wyz@tongji.edu.cn).

MengChu Zhou is with the Macao Institute of Systems Engineering, Macau University of Science and Technology, Macao 999078, China, and also with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: zhou@njit.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TITS.2024.3421894, provided by the authors.

Digital Object Identifier 10.1109/TITS.2024.3421894

shortest path to the final target. Such approaches suit small to medium-sized teams as they consider all environmental viewpoints [22]. However, as the problem scale increases, their performance rapidly declines. The partition-based ones divide environment into sub-areas such that an equal number of robots surveil their respective areas. The cyclic ones are based on graph theory and operational research. The coordinator computes the minimal-cost cycles to visit all viewpoints. Then, robots are assigned along the path and repeat the same surveillance route. The cyclic ones perform well in weakly connected and large-scale environment because of their better use of offline planning [26].

Unmanned ground vehicles (UGVs), viewed as mobile robots, have been increasingly used in road network applications [29], [30], [31], [32] due to their safety, mobility, and efficiency advantages. Persistent Surveillance with UGVs on a road network is a novel research direction recently presented in [33]. It has two single-robot surveillance versions [22], [34]. In this problem, a group of UGVs cooperate to surveil viewpoints on a road network. Unlike previous persistent surveillance problems, it considers the detection ability of UGVs. It allows UGVs to identify a viewpoint's status as long as it is within its detection range. It plans a path with a certain coverage along the network without traversing all the viewpoints. They propose a heuristic cognitive architecture-based path planning method (HCPS), consider the duplication of targets and trajectories among UGVs, and develop three rules for UGVs' decisions. However, HCPS focuses on persistent surveillance without considering emergency handling.

B. Adversarial Surveillance

Persistent surveillance minimizes the visiting time interval to monitor the environment as frequently as possible. This principle leads to deterministic and periodic surveillance routes. Adversarial surveillance focuses on unpredictable movement patterns and prevents intruders from invading, contaminating, and damaging the environment [9]. Adversarial surveillance is widely used in infrastructure guards [35] and perimeter defense [36]. In these strong confrontation scenarios, intruders can decide when and where to attack by analyzing a surveillance strategy adopted by robots [37]. Therefore, adversarial surveillance methods adopt non-deterministic strategies to avoid being easily predicted by intruders.

Sak et al. [38] study adversarial robots on undirected graphs, where they need to ensure that a viewpoint in an environment is not invaded. The probability of detecting an intruder successfully becomes the main evaluation criterion for such problems. The defenders need to maximize the probability for intruders to be detected at viewpoints [37]. A game theory-based approach treats such problems as an adversarial game [39]. A robot performs the optimal strategy by calculating its payoff matrix, which is used for reinforcement learning [40]. Some methods also use time as an evaluation criterion. Alam et al. [41] set the objective of defensive robots to minimize the time between visits to each pair of viewpoints in undirected graphs. Hernandez et al. [35] focus on visiting viewpoints in irregular time intervals. The perimeter defense problem is a typical adversarial surveillance one where robots defend a perimeter of a high-security area [36]. A number of intruders move from the outside towards the perimeter, and a group of defensive robots have to detect the intruders and kill them as they reach the perimeter. To avoid being identified/found by intruders, defenders can move probabilistically, forward and backward, or stand still [42], [43], [44]. This randomness makes intruders unable to predict their trajectory well. Some studies also consider the problem as a task assignment. The location and timing of intruders' appearance on the perimeter are the tasks to assign. Like a perimeter defense problem, it has been systematically studied from single [45] to multiple intruders [46] and linear perimeters [47] to circular areas [48], [49]. The latest work models a perimeter defense problem as a spatiotemporal multitask allocation problem [2], and the use of its proposed method can protect territory with fewer defenders than intruders. Nevertheless, the above studies fail to consider defenders' persistent surveillance when intruders may appear in viewpoints instead of along the perimeter.

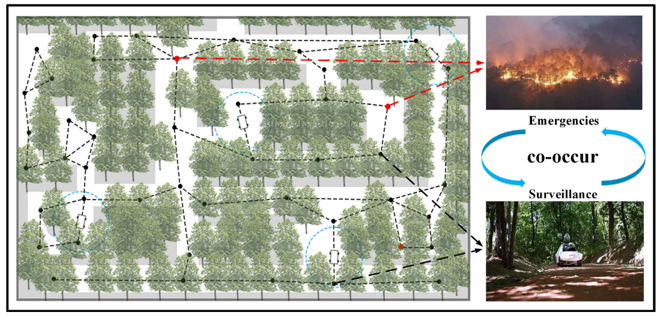


Fig. 1. A forest fire scene for persistent surveillance and emergency handling with multi-UGVs. The black dots and dotted lines represent viewpoints and feasible edges on the road network, and the blue circles represent the detection range of UGVs. They need to surveil these viewpoints to complete persistent surveillance cyclically. The red dots indicate emergencies where fires are present, and UGVs cooperate in prioritizing them.

C. Dual-Task Surveillance

Our literature review suggests that existing methods fail to simultaneously perform persistent surveillance and handle emergencies as dual tasks. According to [9], there is a gap in surveillance studies when a robot considers handling emergencies during its surveillance duty. The dual-task environment in this work is modeled as weighted viewpoints and paths based on the work [33], as shown in Fig. 1. Unmanned ground vehicles (UGVs) surveil in a road network environment such as an urban street, park, and campus. They move along roads, while other areas outside of the roads are obstacles [50]. UGVs move along the paths with sensors to detect multiple viewpoints inside their detection range. Considering the real scene, such as forest fire management [51], terrorist attack [52] and personnel rescue [53], several emergencies may suddenly appear at viewpoints. UGVs surveil viewpoints to detect emergencies and cooperate to recognize them, and then immediately handle them. For example, multiple UGVs work together to extinguish a large fire with their combined water volumes in fire extinguishing. After the emergencies are handled, the viewpoint returns to its normal surveilled status.

In the dual-task surveillance problem, UGVs are not only required to surveil for potential emergencies but also to handle them. Multiple UGVs must handle emergencies cooperatively if a single UGV is not enough for emergencies occurring at a viewpoint. For example, UGVs detect and extinguish a sudden emergent large fire with their equipped water volumes. However, UGVs using existing methods treat emergencies as surveillance. Only if the previous UGV has consumed its equipped item volume and left, other UGVs come one by one to handle the rest. For divisible emergencies that can be handled separately, e.g., litter picking, this leads to a slow response to emergencies. For indivisible emergencies like fires, the existing method may fail to handle them.

In order to enable existing methods to solve the dual-task surveillance problem, the user can divide the emergencies at a viewpoint into several separate surveillance tasks. UGVs can then handle emergencies as surveillance tasks at a single viewpoint. However, the emergent nature of emergencies prevents them from being considered simply as a combination of multiple surveillance tasks since:

1) Competition between surveillance and emergencies may cause UGVs to constantly surveil viewpoints for more rewards, leading to delayed or no handling of emergencies, i.e., emergency starvation. Rewards gained from relatively long-term non-surveilled viewpoints may be greater than or equal to those to handle the emergencies. In this case, if an environmental status changes fast, emergency starvation turns out to be serious because performing surveillance can have higher rewards than handling emergencies. In addition, UGV's surveillance objective function drives them to run more distance to surveil more viewpoints on the path, thus spending more time and causing emergency starvation; and

2) Algorithms can reduce emergency starvation by increasing the rewards of emergencies. However, UGVs lack cooperation. UGVs farther away may take over the tasks of UGVs who are closer to emergencies, causing the delays to handle emergencies. More specifically, to avoid conflicts, when enough UGVs have chosen emergencies as their targets, other UGVs that are closer to the emergencies will not choose them again. Thus, UGVs responsible for the emergencies may not be the closest ones.

Therefore, the methods in the existing literature cannot solve the proposed dual-task problem well. The algorithm for solving the problem needs to avoid emergency starvation and, even more so, drive the most appropriate UGV swarm to handle emergencies cooperatively. This work proposes a cooperative dual-task path planning method to fill this gap. It intends to make the following novel contributions with respect to the state of the art:

1) Proposing and formulating a new dual-task problem in which UGVs must perform both persistent surveillance and emergency handling;
2) Developing a heuristic path planning method capable of effectively solving the dual-task problem. It employs dual-task surveillance priorities, enabling UGVs to handle both tasks without any manual intervention.

Section II reviews the background road network persistent surveillance. A dual-task problem is formulated in Section III. Section IV presents the design of a cooperative dual-task path planning method. The performance comparison between it and the state of the are is given in Section V. Finally, Section VI concludes this article.

TABLE I

All Notations Used In This Work

Symbol	Description	Symbol	Description
G	Road network	R'	Number of UGVs arriving at v_i
v_i	Viewpoint	F_i	Synthetic negative sensing probility
e_i	Edge	Δt_i	Detection time interval
w_i	Weight for viewpoints	t	Time point
p_i	Emergency's position	t_i	Last complete surveillance time
N	Number of viewpoints	$u_i(t)$	Surveillance uncertainty
M	Number of edges	t_n	Time normalizing parameter
K	Number of emergencies	$\Delta \hat{t}_i$	Emergency interval
H_i	Number of emergencies appear at v_i	$\hat{t}_i$	Emergencies' detected time
R	Number of UGVs	$\hat{u}_i(t)$	emergency degree
v	UGV's constant speed	P_i^r	Rewards for viepoints
p_r	Position of rth UGV	O_i	Observable viewpoints
θ_r	Movement direction	α	Emergency's priority
d_t	Time step	p_c	Reward coefficient
f_i	Detection probability	$d_{\min}$	Length of the least-cost path
d_i	Distance between UGV and viewpoint	p_m	Minimum reward coefficient
$\check{d}$	Detection range	p_n	Normal reward coefficient
$\hat{d}$	Observation limit	p_b	Backtrack probability
I_r	UGV's inventory	S_i	Whether emergencies have been handled

II. Dual-Task Problem Formulation

This section defines a dual-task problem for a swarm of UGVs that perform persistent surveillance and emergency handling while moving along the road network. This problem is formulated as a new combinatorial optimization problem based on a road network, UGV model and dual-task effect model. Table I contains all notations used in this work.

A. Road Network Environment

A road network environment is abstracted as a weighted undirected graph $G = (V, E, W, D)$ consisting of N discrete viewpoints and M edges in a two-dimensional region $\Omega \subset \mathbb{R}^2$. $V = \{v_1, \ldots, v_N\}$ is the set of viewpoints surveilled by UGVs. $E = \{e_1, \ldots, e_M\}$ denotes the set of all edges that a UGV can move among viewpoints without turning. $W = \{w_1, \ldots, w_N\}$ is the weight of each viewpoint and represents the importance of viewpoints. $D = \{d_1, \ldots, d_M\}$ is the set of edge lengths, i.e., actual distances between pairs of viewpoints.

B. UGV Model

R homogenous UGVs move along road network G. According to the assumptions in a surveillance problem [9], the speed of each UGV is constant. The physical influences of turning and backtracking are not considered. The size of UGV is negligible compared to the length of an edge. The position of the rth UGV at time t is denoted as $p_r(t) = \left[x_r(t), y_r(t)\right]$, $r \in \{1, 2, \ldots R\}$, following the kinematics:

$$p_r(t + d_t) = p_r(t) + \theta_r(t) \cdot v \cdot d_t \tag{1}$$

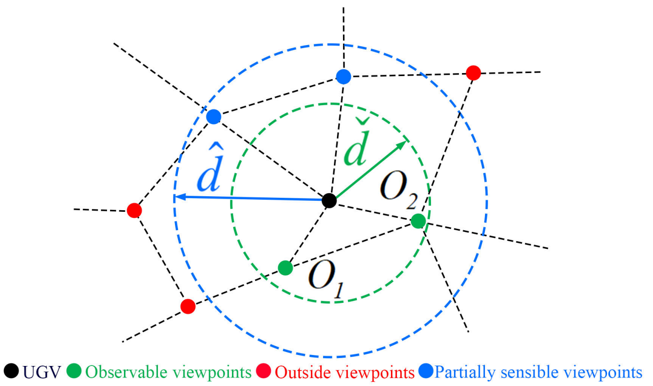


Fig. 2. The black dot is UGV. Green circle is the detection range $\check{d}$. Blue circle is the observation boundary $\hat{d}$. Green dots are viewpoints within $\check{d}$, i.e., observable viewpoints. Blue dots are partially sensible viewpoints between $\check{d}$ and $\hat{d}$. Red dots are outside viewpoints that exceed $\check{d}$ and not detectable. UGV arrives at v_i where it can fully surveil the state of O_1 and O_2. In turn, UGV at O_1 and O_2 can surveil v_i without reaching v_i.

where $p_r(t)$ represents the position of the rth UGV at time t, d_t is a discrete time step, and $\theta_r(t)$ is the vector of a movement direction with $0 \le \theta_r(t) < 2\pi$. v is the constant speed of UGV moving on $\theta_r(t)$ and is a scalar. The 'dot' here means a simple scalar multiplication.

UGV has an omnidirectional 360-degree sensor that can surveil the target's viewpoint status as long as it is within the detection range. This ability is represented as the probability of a viewpoint v_i being accurately surveilled, which is modeled as a piecewise function related to the distance between UGV and its target viewpoint:

$$f_i = \begin{cases} 1, & d_i \le \check{d} \\ \left(\check{d} - d_i\right) / \left(\hat{d} - \check{d}\right) + 1, & \check{d} < d_i \le \hat{d} \\ 0, & d_i > \hat{d} \end{cases} \quad (2)$$

where f_i is the detection probability at v_i, d_i is the Euclidean distance from UGV to v_i, $\check{d}$ is UGV's detection range, and $\hat{d}$ denotes UGV's observation boundary. As shown in Fig. 2, UGV at v_i can surveil surrounding observable viewpoints O_i within $\check{d}$ with a probability of 1. It decreases between $\check{d}$ and $\hat{d}$ and eventually drops to 0 at the observation boundary $\hat{d}$.

Definition 1 (Observable Viewpoints): are a collection of viewpoints within detection range $\check{d}$ from viewpoint v_i.

When multiple UGVs surveil the same viewpoint v_i, the negative probability that all UGVs fail to detect v_i is expressed by the joint probability [54] as

$$F_i = \prod_{r=1}^{R} \left(1 - f_i^r\right) \quad (3)$$

where f_i^r is the detection probability of the rth UGV to v_i. $1 - f_i^r$ is the negative detection probability. They are multiplied to compute the probability that all UGVs do not surveil i. If one UGV's detection range contains v_i, i.e. $\exists f_i^r > 0$, then $F_i < 1$.

The detection time interval Δt_i for viewpoint v_i is described as the time gap between its current visit time t and last surveillance time t_i at v_i:

$$\Delta t_i = t - t_i. \quad (4)$$

UGVs persistently surveil viewpoints' status to reduce their non-confidence levels in a road network over long periods. Thus, its effect needs to reflect the surveillance's dynamic nature and is positively correlated with detection time interval and negatively with detection probability. Based on the above analysis, the uncertainty u_i of viewpoint v_i is used to measure the effectiveness of persistent surveillance and expressed as

$$u_i(t) = \begin{cases} 0, & \exists\, i \in \mathbb{R} \text{ such that } f_{\mathbb{R}} = 1 \\ 1, & u_i(t) > 1 \\ w_i \cdot \Delta t_i \cdot F_i / t_n, & \text{otherwise} \end{cases} \quad (5)$$

where w_i is the importance weight of v_i. t_n is an expectation interval to normalize the uncertainty as a dimensionless variable and can be estimated by $\max w_i \cdot \Delta t_i \cdot F_i$. When t_n is relatively small, the uncertainty grows fast with time. Considering the real scene, a viewpoint's maximum allowable interval Δt_i between two visits should be constrained according to the environment types, such as 1 hour for warehouse monitoring. Otherwise, the system fails to detect events appearing at v_i. $u_i(t) = 0$ only if $f_{\mathbb{R}} = 1$, which means that the viewpoint v_i is within the detection range of at least one UGV, and $\Delta t_i = 0$.

C. Dual-Task Effect Model

In the road network environment, dual tasks may arise at the viewpoints. A UGV can detect viewpoints' status in its detection range; while emergencies can only be handled when UGVs reach the viewpoints according to realistic scenarios like catching criminals and extinguishing a fire. $P = \{p_1, \ldots, p_K\}$ is a set of viewpoints where emergencies appear, and K is the number of emergencies. The road network environment with emergencies is abstracted as $G = (V, E, W, D, P)$.

H_i is the number of emergencies co-appearing at the viewpoint v_i, and is denoted as

$$H_i = \sum_{j=1}^{K} [p_j = v_i] \quad (6)$$

where $[x] = 1$ when condition x is satisfied; otherwise 0.

UGVs have equipment capable of handling emergencies in a road network G. Once emergencies appear, UGVs detect and share their positions, then arrive at the position to handle the emergency. Emergencies are handled once necessary UGVs have reached. UGVs have limit to handle emergencies, meaning that a UGV can only use its equipment to cope with one emergency. After its equipment is used, it cannot handle a new emergency, which is described as:

$$I_r = \begin{cases} 0, & \text{no equipment} \\ 1, & \text{otherwise} \end{cases} \quad (7)$$

where I_r denotes whether the rth UGV can solve an emergency.

Considering realistic scenarios, emergencies may appear at the same viewpoint v_i, i.e., $H_i > 1$, and are divided into two categories. One is indivisible emergencies such as forest fires [51] where multiple UGVs need to meet at the target viewpoint to deal with the fire simultaneously. Otherwise, for example, if a UGV first puts out part of the fire, the fire may soon grow up.

$$S_i = [\sum_{r=1}^{R'} I_r \geq H_i] \tag{8}$$

where S_i defines whether H_i indivisible emergencies have been handled at v_i, and R' is the number of UGVs arriving at v_i. $[x] = 1$ when condition x is satisfied; otherwise 0. $S_i = 1$ if enough UGVs whose $I_r = 1$ arrive at emergency locations, meaning that these indivisible emergencies are handled. When the number of UGVs arriving at emergencies is less than H_i, emergencies cannot be well handled, and thus $S_i = 0$.

The other one is divisible emergencies such as litter picking in public places [55], [56]. When a UGV detects such litter, it fills its bin as much as possible and leaves without waiting to meet other UGVs. Other UGVs arrive at different moments to pick up the remaining litter. A divisible emergency that appears at v_i can be resolved if UGV r has passed through it.

UGVs aim to immediately handle emergencies at viewpoint v_i in the dual-task problem. Thus, the evaluation criterion is modeled as the emergency degree and needs to consider the number and type of emergencies, which are expressed as:

$$\hat{u}_i(t) = (1 - S_i) \cdot H_i \tag{9}$$

where $\hat{u}_i(t)$ denotes the emergency degree of viewpoint v_i at time t. $\hat{u}_i(t)$ is positively related to the number of emergencies as H_i. For indivisible emergencies, $\hat{u}_i(t)$ equals 0 or H_i, depending on whether all needed UGVs arrive at v_i, i.,e., $S_i =$ 0 or 1. For divisible emergencies, $S_i = 0$, and $\hat{u}_i(t) = H_i$.

The performance of the proposed dual-task problem is measured by the risk level:

$$C = \frac{1}{N}\sum_{i=1}^{N}(u_i(t)) + \sum_{i=1}^{N}\hat{u}_i(t). \tag{10}$$

D. Dual-Task Problem Formulation

A dual-task problem is defined as: given R homogenous UGVs, the sensing ability f_i, number of emergencies H_i at v_i, a road network $G = (V, E, W, P)$, and UGV moving speed v, UGVs plan their paths to minimize the uncertainty and emergency degree at all viewpoints. It is assumed that the uncertainty of all viewpoints at the beginning of the task is 1 and the emergency degree is 0. The dual-task problem is formulated as follows:

$$f = \arg\min_{\mathrm{T}}(C) \tag{11}$$

by finding a group of trajectories $T = [T_1, \ldots, T_r, \ldots, T_R]$ of R UGVs to minimize C, where T_r is the trajectory of the rth UGV and represented by a sequence of viewpoints traversed by it, i.e., $T_r = (v_{\mathrm{a}}, v_{\mathrm{b}}, \ldots)$, where $v_{\mathrm{a}}, v_{\mathrm{b}}, \ldots \in V$, $1 \leq r \leq R$, subject to (1), (2) and (7).

The surveillance problem has been proved NP-hard by reducing it to a traveling salesman problem [33]. The proposed dual-task surveillance problem includes surveillance problems and thus NP-hard. Theoretically, there is no optimal solution of polynomial time complexity for an NP-hard problem. Research shows that the heuristic method is an effective means for solving the NP-hard problem [21]. It can be combined with online planning to handle dual-task surveillance timely. Thus, we propose a rule and emergence mechanisms-based heuristic method to solve the new problem.

III. Cooperative Dual-Task Path Planning

The cognitive architecture [57] is widely used in solving a surveillance problem. There is a centralized coordinator planning on the global graph to decide which vertex to move to. It knows all viewpoint statuses and assigns the target viewpoint with the highest reward for robots. Each robot then chooses the shortest path to its assigned targets. This architecture allows UGVs to consider all viewpoints and thus has an advantage in finding global optimal paths. Considering the advantages of cognitive architecture in dynamic planning [57], we propose a Cooperative Dual-task Path planning method (CDP) to solve the dual-task problem based on it. Its basic process for a UGV is: 1) Select all the viewpoints as candidates. 2) Compute candidates' rewards with their risk level and update the shortest path from the UGV to them. 3) If the UGV is not in the closest swarm to a candidate with emergencies, clear its reward such that it will not choose the emergencies as its target. 4) Select the UGV's target viewpoint with the maximum reward. 5) The UGV moves one step to approach its target viewpoint according to the shortest path. 6) UGVs replan their targets and paths when they reach a viewpoint. The proposed method considers all the viewpoints in the graph, which is widely used in dynamic path planning [57].

A. Target Viewpoint With Dual-Task Priorities

UGVs first select a target viewpoint. Determining a target viewpoint requires the consideration of three factors: the reduction of uncertainty, the resolution of emergencies, and the moving time consumption [16]. Dual-task priority is applied in calculating the decision reward to ensure that the highest priority is given to emergencies. The combined reward is built as follows:

$$P_i^r = \sum_{k \in O_i} u_k(t) + \alpha \cdot I_r \cdot \hat{u}_i(t) - p_{\mathrm{c}} \cdot d_{\min}/v \cdot w_i \tag{12}$$

where P_i^r represents the reward for UGV r selecting the viewpoint v_i as its target. O_i is the observable viewpoints [33] and defined in Definition 1, $u_k(t)$ is the uncertainty of the viewpoint v_k belonging to O_i; α is the dual-task priority weight to ensure that the emergency has a higher priority. p_c is a reward coefficient for the path's cost in calculating a viewpoint's reward. $d_{\min}$ is the shortest path distance. When the target viewpoint v_t has emergencies, the shortest path's destination is exactly v_t. However, when there is no emergency, the destination is one of v_t's observable viewpoints O_t because a UGV only needs to reach one of O_t that can surveil v_t. The shortest path between any two viewpoints is calculated using the Floyd-Warshall algorithm [58]. It considers a graph G with

Algorithm 1 Computing Observable Viewpoints and α

Input: The set of viewpoints V, detection range $\check{d}, k = 0$
Output: the set of observable viewpoints O_i, α

for each viewpoint v_i **do**
    for each viewpoint v_j **do**
        $d_{ij} \leftarrow$ distance between v_i and v_j;
        if $d_{ij} \leq \check{d}$ **then**
            $O_{ik} = v_j$;
            $k = k + 1$;
        end
    end
    if $k + 1 > \alpha$ **then**
        $\alpha = k + 1$;
    end
    $k = 0$.
end

Algorithm 2 Computing the Closest Swarm of UGVs

Input: Current target viewpoints of other UGVs Q, $k = 0$, $P_{\min} = \infty$, waitinglist = 0
Output: the closest swarm of UGVs

$Q \leftarrow$ current target viewpoints of other UGVs; $k = 0$, $P_{\min} = \infty$, waitinglist = 0 ;
for each viewpoint v_i **do**
    Clear waitinglist;
    for r'th UGV's target $v_j \in Q$ **do**
        if $v_i = v_j$ **then**
            Compute $P_j^{r'}$;
            $k = k + 1$;
            Add $(r', P_j^{r'})$ to waitinglist;
        end
    end
    if $k = H_i$ **then**
        Sort waitinglist by reward;
        $(r', P_j^{r'}) \leftarrow H_i$ th element in waitinglist;
        if $P_i^r > P_i^{r'}$ **then**
            Remove v_i from UGV r''s target;
        end
        else
            $P_i^r = 0$.
        end
    end
end

viewpoints V numbered 1 through N and defines a function $F(i, j, K)$ that returns the shortest path length from v_i to v_j via intermediate viewpoints $v_k \in \{v_1 \cdots v_K\}$. When all the possible $v_k \in \{v_1 \cdots v_N\}$ are considered, the shortest path with multiple edges is updated and obtained according to the recursive formula: $\mathrm{F}(i, j, K) = \min(\mathrm{F}(i, j, K-1), \mathrm{F}(i, K, K-1) + \mathrm{F}(K, j, K-1))$. The calculation processes of the observable viewpoints and α are given in Algorithm 1.

When p_c is large, UGV mainly considers moving short distances to maximize the reward. Therefore, closer viewpoints will be more likely to be selected. When p_c is small, UGV focuses more on the risk level of the global viewpoint. According to the analysis in [33], the reward coefficient represents the degree of adjustment to the global and local road network persistent surveillance performance. Therefore, an adaptive adjustment function is similarly implemented for p_c to optimize the performance of the dual-task path planning.

$$p_{\mathrm{c}} = \begin{cases} p_{\mathrm{n}}, & u_{\mathrm{m}} < u_{\max} \wedge H_i = 0 \\ p_{\mathrm{m}}, & u_{\mathrm{m}} \geq u_{\max} \vee H_i > 0 \end{cases} \tag{13}$$

where p_m and p_n are respectively the minimum and normal reward coefficients, $u_{\max}$ is the allowable uncertainty, and u_m is the current maximum uncertainty. When the maximum uncertainty in the environment is greater than u_m, or UGVs discover emergencies, the weight of path cost decreases from p_n to p_m. The algorithm drives UGVs to the viewpoint with the maximum risk level. There is $p_m < p_n < 1$, and $p_m \to 0$, $p_n \leq N/t_n$.

B. Closest Swarm of UGVs to the Target Viewpoint With Emergencies

Multiple UGVs may select the same target viewpoint and cause a waste of surveillance performance. Inspired by [33], a location dispersion rule is utilized to resolve the target conflict during persistent surveillance: All viewpoints within $\check{d}$ around a viewpoint that the UGV has selected have a reward of 0 to the other UGVs. However, this rule fails to consider emergencies because emergencies appearing at a viewpoint may require more than one UGV to handle. A heuristic protocol is employed for each UGV to compute a swarm of the nearest UGVs to handle the emergencies. UGVs compete for target viewpoints and finally find the closet subswarm by reward calculation. This protocol is implemented in Algorithm 2 and its output is the closest swarm.

A UGV first obtains target viewpoints $v_j \in Q$ of other UGVs, calculates the times k that its target viewpoint is the same with others, and adds the record to the waiting list, which stores the index of the UGV with the repeated target and its reward. If $k = H_i$, the currently selected viewpoint v_i has been allocated enough UGVs to handle it. Selecting this viewpoint as the current UGV's target will create a new conflict, and the UGV needs to compete with the r'th UGV, which has the lowest reward in the waiting list. If the competition fails, then $P_i^r = 0$; Otherwise, v_i is removed for r' and it will reselect the target viewpoint in the next decision. When all UGVs have calculated their viewpoints' rewards in a time step, a subswarm of UGVs with the closest distance actively select the target viewpoint and handles emergencies.

C. Next Viewpoint to Visit

After UGVs select the target viewpoint and optimize the selection result by Algorithm 2, they compute the shortest path to the target viewpoint and decide which viewpoint to visit. Considering the environment's dynamics in uncertainty and urgent degree, a UGV moves one step along the shortest path. It recalculates its target viewpoint and shortest path when

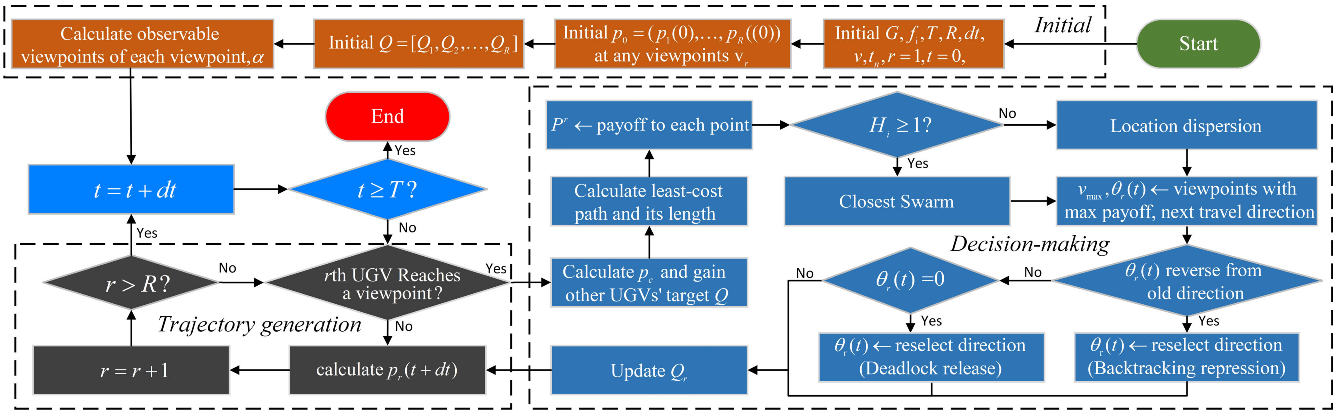


Fig. 3. Flowchart of the proposed CDP. Q_r represents the target viewpoints of the rth UGV.

Algorithm 3 Cooperative Dual-Task Path Planning

Input: Task duration T_d, number of UGVs R, time step dt, v,t_n,initial positions p_0
Output: Trajectories of UGV r

1 **Load:** Undirected graph $G=(V,E,D,W,P)$, function f_p; Observable viewpoints of each UGV r, α; **(Algorithm 1)**
2 **for** each t in T_d at intervals of dt **do**
3 $\quad u_i(t), \hat{u}_i(t) \leftarrow$ update uncertainty and emergency degree of each viewpoint using (5,9);
4 $\quad$ **if** the rth UGV arrive at a viewpoint **then**
5 $\quad\quad$ Compute p_c through (13);
6 $\quad\quad$ Gain other UGVs' target;
7 $\quad\quad$ Compute least-cost path and its length;
8 $\quad\quad$ Compute P_i^r through (12);
9 $\quad\quad$ **if** $H_i \geq 1$ **then**
10 $\quad\quad\quad$ Computing the closest swarm; **(Algorithm 2)**
11 $\quad\quad$ **else**
12 $\quad\quad\quad$ Location dispersion;
13 $\quad\quad$ $v_{\max} \leftarrow$ find(max P_i^r)
14 $\quad\quad$ $\theta_r(t) \leftarrow$travel direction to $v_{\max}$
15 $\quad\quad$ **if** $\theta_r(t)$ reverse from old direction **then**
16 $\quad\quad\quad$ Backtracking repression;
17 $\quad\quad$ **else if** $\theta_r(t)=0$ **then**
18 $\quad\quad\quad$ Deadlock release;
19 $\quad$ Compute $p_r(t+dt)$ using (1).

---

reaching the next viewpoint. During this period, two rules are considered as follows.

1) Deadlock release: When a UGV always stays at a viewpoint, it is defined as a deadlock. It occurs when the path cost is greater than the reward or when multiple viewpoints have equal rewards. To release UGVs from a deadlock, the adjacent viewpoint with the maximum reward is selected as the target viewpoint for the next step.

2) Backtracking repression: Due to the dynamical uncertainty of the viewpoint state, a UGV is likely to move back and forth on the road network. When UGV's moving direction is reversed from the old direction, the backtracking occurs and the UGV will backtrack with a probability, denoted as p_b:

$$p_{\mathrm{b}}=\begin{cases}1, & d_{\mathrm{g}}=1 \vee H_i>0\\ 0.4/d_{\mathrm{g}}, & d_{\mathrm{g}}>1 \vee I_r=0\end{cases} \tag{14}$$

where d_g is the edge degree of the viewpoint v_i. When UGV has only one direction to move or needs to handle emergencies, it is unavoidable to backtrack. Otherwise, it will choose the adjacent viewpoint with the highest reward as the next target viewpoint with probability. The probability parameter setting of 0.4 is according to [33].

If a UGV arrives at a viewpoint with indivisible emergencies, it waits there for the arrival of other UGVs. Otherwise, it continues to surveil viewpoints.

Fig. 3 and Algorithm 3 show the proposed method's flowchart and pseudocode, called Cooperative Dual-task Path planning (CDP). CDP starts with the initialization of a map and UGVs. Then, CDP computes targets for UGVs to generate trajectories by the following steps: At each time step t, UGV determines whether it has reached a viewpoint and, if so, it makes a new decision: UGV calculates reward coefficient p_c and gains other UGVs' targets to compute viewpoints' reward. Considering UGVs' cooperation, UGV chooses the closest swarm or location dispersion to resolve target conflicts according to whether it needs to handle emergencies. After reward computation, UGV selects the direction of the shortest path to the viewpoint with the maximum reward. If the selected direction triggers a backtrack or deadlock, it selects a new direction to repress the backtrack or get released. Finally, it moves to the next viewpoint in the planned direction. Its complexity is determined by steps 7 and 8, where the shortest path and reward are calculated. The complexity of finding the shortest path is $O(N^2)$, as the Floyd-Warshall algorithm is employed. Considering the number of observable viewpoints of viewpoint v_i as n_i, the shortest path needs to be calculated $N \cdot n_i$ times. Therefore, the worst complexity of CDP is $O(\max(n_i)\cdot N^3) \ll O(N^4)$.

The proposed algorithm CDP has two types of parameters. The first type includes environmental and UGV model parameters such as map, UAV speed and detection range. They do not affect the ability of CDP. The other type includes those directly affecting the cooperation among UGVs, e.g., UGV's cooperation range d'; all viewpoints within d' around a viewpoint that the UGV has selected have a reward of 0 to the other UGVs. A reasonable setting for d' can improve UGV cooperation efficiency and make the algorithm find better solutions that can be close to the optimal solution (no guarantee since the problem is NP-hard). Our deeper experiments and analysis indicates that when the optimal cooperation range d' equals UGV's observation boundary $\hat{d}$, the proposed algorithm performs relatively the best. Relative materials are given in Section X of the Supplementary File.

IV. Simulation Results

The performance of the proposed CDP is tested by simulation using various road network models, detection range, dual-task priority weight α and emergency types to illustrate its feasibility. The state-of-art surveillance method is compared in terms of the method's accuracy for solving the dual-task problem with various UGV numbers and emergencies' appearing cycles. The simulation results and analysis demonstrate the proposed method's superiority in solving the dual-task problem.

A. Scenes and Performance Criteria

As shown in Fig. 4, the experiment considers three different road network environments, marked as M_1, M_2, and M_3. M_1 is a small-scale map with strong connectivity. It contains 49 viewpoints, which are evenly distributed, $10m$ apart, covering an area of $3600m^2$. M_2 simulates a real-world irregular road network and contains 64 viewpoints, covering an area of $4940m^2$. M_3 is a large-scale map with several obstacles, containing 263 viewpoints, covering an area of $26600m^2$. The viewpoints in all the maps are numbered sequentially from top to bottom and from left to right. The blank in M_1 and M_2 is the same as the obstacles in M_3, where UGV cannot reach.

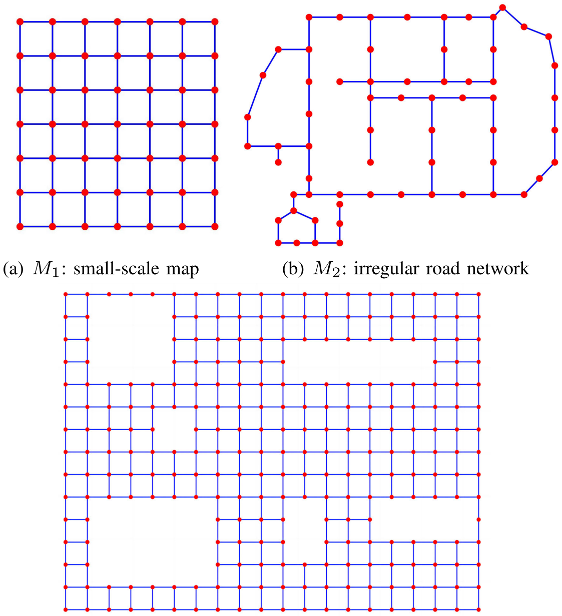
Emergencies appear cyclically and randomly in the environment with a cycle of t_f and a UGV's equipment I_r is refilled in every cycle. In M_1 and M_2, for every $t_f = 200s$, there are 2 viewpoints each with 2 emergencies appear. In M_3, for every $t_f = 200s$, there are 2 viewpoints each with 3 emergencies appear. The divisible emergency type is marked as E_1 and the indivisible emergency type is marked as E_2.

All simulations default conditions are under E_1, $\hat{d} = 2\check{d}$, $d_t = 1s$, $T_d = 1200s$. $v = 1m/s$, 4 UGVs in M_1 and M_2, $v = 10m/s$, 6 UGVs in M_3, and $t_n = 1000s$. The other parameters are $u_{\max} = 0.80$, $p_n = 0.01$ and $p_m = 0.001$. All the viewpoints' weights are 1.

The method's performance for solving the dual-task problem is measured according to the average risk level (C_1) and the successfully handled emergencies ratio (C_2). C_1 is calculated by the formula (11). C_2 is the ratio of successfully handled emergencies' number H_s to the total emergencies' number H_t. The calculation formula is $C_2 = H_s/H_t$.

B. Results and Analysis

The feasibility of the proposed CDP is first illustrated in the following contexts. Then the accuracy is compared with the state-of-the-art road network persistent surveillance method [33] and its modified variation. All the simulation



(a) M_1: small-scale map (b) M_2: irregular road network

(c) M_3: large-scale map

Fig. 4. Three different road network environments. The red dots represent the viewpoints and the blue segments represent the edges.

results in this section are obtained from MATLAB 2018b on a desktop computer with an Intel Core i7-10750H (2.6 GHz), Corsair DDR4 (16GB) memory, and Nvidia Geforce GTX 1650Ti (4GB) graphics card. Considering the randomness of the heuristic method, the performance criteria $C1$ and C_2 in this section are the averages of consecutive calculations.

1) Feasibility: The feasibility means that UGVs can surveil all viewpoints' statuses and simultaneously handle emergencies. Figs. 5 and 6 show the behavior of UGVs planned by the proposed CDP and the surveillance effect qualitatively. As shown in Fig. 5, UGVs cooperatively surveil all viewpoints and handle emergencies in the road network environment. Since the detection range of UGV is considered, when some viewpoints are less than the detection range from their neighbors, UGV can detect the status of all viewpoints without traversing them.

As shown in Fig. 6, The variation of the average risk level further illustrates the execution of the method. In the beginning, UGVs persistently surveil viewpoints, causing the risk level to decrease. When emergencies appear, the risk level suddenly increases. UGVs first detect emergencies and then travel to handle them with higher priority. When emergencies are handled, the risk level falls to a low level and UGVs continue to surveil viewpoints until new emergencies appear. As the detection range increases, the risk level decreases in a shorter time because emergencies can be detected by UGVs more quickly.

As shown in Table I, different dual-task priorities affect the method's performance in handling dual tasks. In the results under $\alpha = 1, 3$, the environment is high-risk and UGV cannot handle all emergencies. This is because a set of viewpoints with high uncertainty probably has a greater reward than handling emergencies at a viewpoint. Thus, UGVs focus on reducing the viewpoint's uncertainty and ignoring emergencies. It is considered that the dual-task priority has a lower bound $\alpha = \max|O_i| + 1$, which means even if all observable

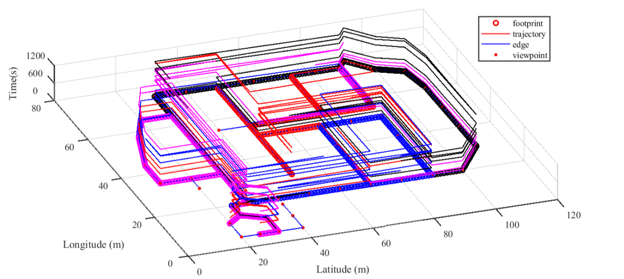


Fig. 5. The trajectories of 4 UGVs under E1, M_2, $\check{d}$ = 10 cm. Different color trajectories represent different UGVs.

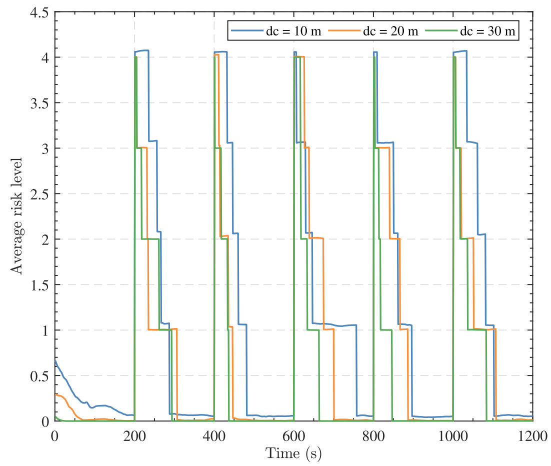


Fig. 6. Average risk level under different detection ranges. All the above results are obtained under conditions of E_1 in M_2.

TABLE II

DIFFERENT DUAL-TASK PRIORITY RESULTS

Dual-task priority	C_1	C_2
$\alpha = 1$	2.92	0.33
$\alpha = 3$	2.12	0.83
$\alpha = 6$	1.56	1
$\alpha = 9$	1.62	1

TABLE III

DIFFERENT EMERGENCY TYPES RESULTS (C_1/C_2)

Emergency type	Divisible emergency (E_1)	Indivisible emergency (E_2)
G_1	1.35/1	0.79/1
G_2	1.81/1	1.45/1
G_3	1.23/1	0.99/1

viewpoints around a viewpoint have an uncertainty of 1, UGVs still prioritize emergencies. Map M_2 has max $|O_i| = 5$, so in the results under $\alpha = 6, 9$, UGVs can prioritize all emergencies and keep the environment at a low risk level.

Table III and Fig. 7 show the performance of the proposed CDP for different emergency types. All emergencies can be handled, and a low risk level is maintained in three maps. Compared to the risk level convergence curve for divisible emergencies in Fig. 6, Fig. 7 remains at a level (e.g., 2, 4, 6) for a longer time because UGVs need to wait for other UGVs to handle emergencies. In contrast, divisible emergencies do not, and UGVs can continue to surveil viewpoints. Thus, divisible emergencies have a lower risk level than indivisible ones, as shown in Table III. The above results show that the method is feasible for solving the dual-task problem.

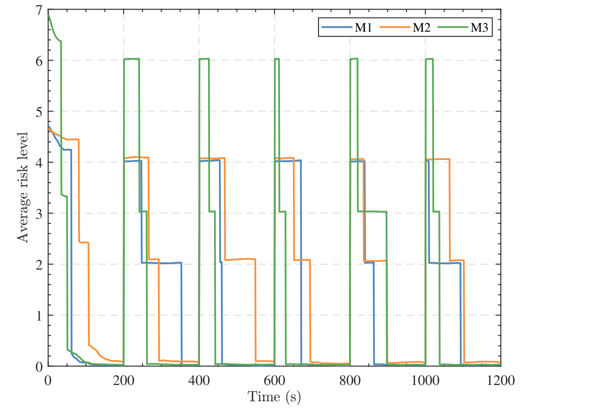


Fig. 7. Average risk level in three maps under conditions in indivisible emergencies E_2.

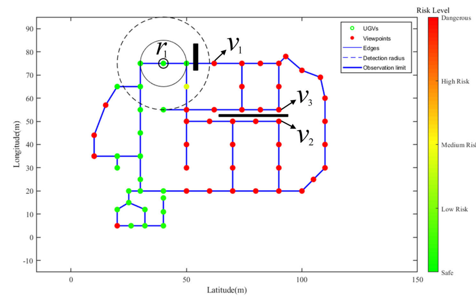


Fig. 8. 4 UGVs represented with different colors are surveilling a road network with dynamic obstacles. The black squares represent obstacles. The viewpoints' color indicates their risk level.

CDP can successfully address dynamic obstacles: 1) UGVs cannot pass through obstacles. 2) UGVs cannot surveil viewpoints obscured by obstacles. For the first case, CDP's UGVs in the last version can directly update the road network map and re-plan paths to avoid obstacles. As shown in Fig. 8, a dynamic obstacle stops the passage of UGV r_1 to its target v_1. Then, CDP drives r_1 to recompute v_1's reward without considering the path having an obstacle. Thus, r_1 does not pass through obstacles and reselect targets for a higher reward because a longer path to avoid obstacles reduces v_1's reward. For the second case, CDP's UGVs compute rewards considering obstacles. As shown in Fig. 8, a dynamic obstacle exists between v_2 and v_3. When a UGV selects v_2 as a target, it does not compute the reward of surveilling v_3 within its detection range. It cannot pass a road to surveil v_2 by arriving at v_3. Thus, surveilling v_2's reward is low, and UGVs prioritize to surveil areas without obstacles. Besides, based on their sensor model, UGVs do not consider viewpoints obscured by obstacles. The experimental results are given in the Supplementary File.

Considering practical applications, more simulation results and analyses are given in the Supplementary File, including the optimal number of robots, homogeneous UGVs, UGV model assumptions, fuel level, sensing failures, noise, and transmission delays.

2) Accuracy: The accuracy means the proposed CDP can handle the dual-task problem with minimum average risk level and maximum success in handling emergencies. Our CDP has been compared with six other state-of-the-art surveillance algorithms, including the heuristic cognitive persistent surveillance algorithm (HCPS) [33], the heuristic pathfinder

TABLE IV

THE PROPOSED METHOD COMPARED WITH OTHER SURVEILLANCE SOLUTIONS IN VARIOUS UGV NUMBERS (C_1/C_2)

	UGV number	$R=1$	$R=2$	$R=3$	$R=4$	$R=5$	$R=6$
M_1	HCR	3.22/0.25	2.64/0.48	1.85/0.71	1.60/0.78	1.40/0.96	1.30/0.96
	HCR-m	3.21/0.25	2.35/0.50	1.71/0.73	1.37/0.84	1.07/0.96	1.04/0.98
	HPCC	3.21/0.25	2.28/0.50	1.69/0.75	1.07/1	0.80/1	0.88/1
	HPCC-m	3.21/0.25	2.20/0.50	1.67/0.75	0.76/1	0.56/1	0.61/1
	HCPS	3.23/0.25	2.32/0.50	1.62/0.75	1.00/0.92	0.83/0.98	0.73/1
	HCPS-m	3.21/0.25	2.21/0.50	1.64/0.75	0.70/1	0.55/1	0.51/1
	CDP	3.21/0.25	**2.16/0.50**	**1.55/0.75**	**0.54/1**	**0.45/1**	**0.31/1**
M_2	HCR	3.80/0.21	2.99/0.43	2.29/0.68	1.94/0.78	1.88/0.78	1.72/0.87
	HCR-m	3.79/0.22	2.82/0.48	2.25/0.68	1.72/0.89	1.51/0.90	1.33/0.96
	HPCC	3.73/0.21	2.84/0.49	2.15/0.72	1.80/0.98	1.45/1	1.34/1
	HPCC-m	3.67/0.23	2.58/0.50	2.11/0.72	1.51/0.95	1.25/1	1.19/1
	HCPS	3.58/0.25	2.83/0.49	2.09/0.72	1.76/0.94	1.43/0.97	1.27/1
	HCPS-m	3.68/0.22	2.64/0.50	2.09/0.75	1.43/1	1.17/1	1.05/1
	CDP	3.68/0.23	**2.55/0.50**	**1.84/0.75**	**1.08/1**	**0.96/1**	**0.75/1**
M_3	HCR	5.24/0.17	4.73/0.29	4.33/0.38	3.85/0.50	3.42/0.59	3.26/0.62
	HCR-m	5.24/0.17	4.60/0.32	4.12/0.45	3.54/0.56	3.35/0.65	3.13/0.71
	HPCC	5.29/0.17	4.32/0.33	3.34/0.50	2.56/0.67	2.04/0.83	1.08/1
	HPCC-m	5.25/0.17	4.29/0.33	3.19/0.50	2.48/0.67	1.98/0.83	0.94/1
	HCPS	5.21/0.17	4.18/0.33	3.36/0.50	2.64/0.67	1.85/0.83	1.18/1
	HCPS-m	5.18/0.17	4.12/0.33	3.16/0.50	2.45/0.67	1.69/0.83	0.83/1
	CDP	**5.12/0.17**	**4.00/0.33**	**3.03/0.50**	**2.37/0.67**	**1.54/0.83**	**0.69/1**

conscientious cognitive (HPCC) [57] and the heuristic conscientious reactive (HCR) [26]. The other three methods are the modifications of HPCC, HCR and HCPS for dual tasks, respectively, named HPCC-m, HCR-m and HCPS-m. HCPS allows UGVs to identify a viewpoint's status as long as it is within the detection range and plans the path with certain coverage along the network without traversing all the viewpoints. HPCC and HCR are recognized as efficient and classical algorithms in the field of UGV surveillance. When solving the dual task problem, UGVs of these algorithms treat emergencies as surveillance tasks and occupy a unique viewpoint for surveillance. However, the viewpoint where the emergencies appear should be shared rather than unique such that multiple UGVs can handle the emergencies quickly. Therefore, the comparison with these original algorithms cannot sufficiently illustrate the advantages of our proposed algorithm, CDP, in the cooperative handling of emergencies. For this reason, we convert emergencies of HCR-m, HPCC-m and HCPS-m into shared surveillance tasks for UGVs.

Table IV compares the experimental comparison results of proposed algorithm CDP to six surveillance algorithms (HCR, HCR-m, HPCC, HPCC-m, HCPS, and HCPS-m) in various UGV numbers and maps. C_1 and C_2 are performance criteria that indicate the average handling level of dual tasks and UGVs' ability to respond to emergencies; Low C_1 means that the dual-task surveillance is handled well and the environment is safe; $C_2 = 1$ means that the emergencies are all handled. Their definitions are given in Sec.IV-A.

Firstly, comparison results with the algorithm X $\in$ {HCPS, HPCC, HCR} and its variant X-m illustrate that the dual task problem is new and differs from classical surveillance in that multiple UGVs are required to handle emergencies. X treats emergencies as unique surveillance. UGVs need to come one by one, and each can only handle a part of the emergencies, leading to a slow response. However, UGVs of X-m share their surveillance targets and can travel to emergencies simultaneously for quick handling. Thus, the performance of algorithms X is weaker than their modified variants X-m. Secondly, the comparison results between the algorithm X-m and the proposed algorithm CDP emphasize the advantages of our algorithms over other solutions: 1) HCPS-m does not handle two types of competition: competition between surveillance and emergencies and competition among UGVs, leading to delayed or non-handling of emergencies. In contrast, our CDP provides a dual-task priority weight α in Algorithm 1 to ensure that the emergency has a higher priority than surveillance. CDP also selects a necessary and appropriate subswarm of UGVs with the nearest distance to handle emergencies instead of just sharing the emergencies, since the more distant UGVs may take over the tasks of UGVs who are closer to emergencies. 2) HPCC-m drives UGVs to arrive at every viewpoint for surveillance and emergency handling. However, our CDP considers UGV's detection range, meaning that UGVs can surveil a viewpoint at its near viewpoints. CDP plans paths for UGVs to traverse only partial viewpoints, thus saving traveling time. 3) HCR-m selects only neighboring viewpoints as path-planning targets. Therefore, UGVs only handle emergencies appearing at their neighbors, making the emergencies easily delayed and thus less effective than our CDP. Thirdly, the comparison results of different maps and UGV numbers indicate that our CDP is more suitable for complex scenarios with more emergencies and big UGV numbers. When $R = 1$, the performance is similar among the six methods. As the number of UGVs increases, CDP has the lowest risk level and can handle all emergencies. Besides, the advantage of CDP is obvious in the irregular map M_2 and the large map M_3. This is because more UGVs and complex environments require more effective UGV swarm cooperation strategies. In summary, the accuracy of our CDP is best with the lowest C_1 and highest C_2. With the increased number of UGVs and more complex maps, CDP's accuracy advantage is more obvious. The above results are obtained under $T_d = 3600s$ and $t_f = 300s$.

TABLE V

THE PROPOSED METHOD COMPARED WITH OTHER METHODS IN VARIOUS EMERGENCIES' APPEARING CYCLE (C_1/C_2)

t_f	300s	250s	200s	150s	100s
HCR	1.94/0.78	2.33/0.68	2.53/0.69	2.86/0.57	3.32/0.41
HCR-m	1.72/0.89	2.04/0.75	2.29/0.71	2.70/0.61	3.23/0.44
HPCC	1.80/0.98	2.00/0.87	2.41/0.72	2.64/0.62	3.22/0.44
HPCC-m	1.51/0.95	1.79/0.91	2.25/0.89	2.54/0.71	3.15/0.54
HCPS	1.76/0.94	1.88/0.82	2.36/0.78	2.68/0.69	3.36/0.4
HCPS-m	1.43/1	1.58/0.95	2.28/0.87	2.48/0.83	3.19/0.51
CDP	**1.08/1**	**1.44/1**	**1.65/0.96**	**2.10/0.87**	**2.71/0.69**

Comparison results under shorter emergencies appearing cycles better indicate our CDP's accuracy advantages for solving dual tasks. Table V presents the proposed CDP compared with other methods under various emergences' appearing cycles t_f in map M_2. When $t_f \geq 250s$, only our CDP has $C_2 = 1$, which means that only CDP can handle all the emergencies within the cycle. When $t_f < 250s$, emergencies appear so fast that UGVs of all the methods do not have enough time to handle emergencies, thus $C_2 < 1$. In these extreme environments, our CDP still performs the best. It handles at least 15 percent more emergencies than the six compared methods, and has the lowest average risk level C_1.

V. Conclusion

This work has investigated a new problem of simultaneously performing persistent surveillance and emergency handling by detection-capable UGVs on a road network, which is named a dual-task problem. We present a novel method to solve it. The simulation results demonstrate its feasibility and advantages in surveilling viewpoints and handling emergencies.

The CDP proposed in this paper needs UGVs' global communication and pre-known maps. It is thus limited in the case of no communication and unknown environments. A reasonable solution to the communication constraint is to establish protocols for UGVs to meet periodically to exchange information, or to have specific UGVs traversing among UGVs as messengers. A viewpoint buffer and task assignment mechanism can be set up for unknown maps to expand the viewpoints set each UGV is responsible for dynamically. When UGVs traverse all viewpoints once, the map is known, and the presented method can work again. Our future work tends to focus on dynamic road network environments, such as dynamic viewpoints and obstacles, which pose new challenges to the method. How to use recent methods [59], [60], [61], [62] to handle a large-scale dual-task problem and more application scenarios [63], [64], [65] should also be investigated.

References

[1] H. Kim, J. Ben-Othman, K.-I. Hwang, and B. Choi, "Intelligent aerial-ground surveillance and epidemic prevention with discriminative public and private services," *IEEE Netw.*, vol. 36, no. 3, pp. 40–46, May/Jun. 2022.

[2] S. Velhal, S. Sundaram, and N. Sundararajan, "A decentralized multirobot spatiotemporal multitask assignment approach for perimeter defense," *IEEE Trans. Robot.*, vol. 38, no. 5, pp. 3085–3096, Oct. 2022.

[3] I. Maza, F. Caballero, J. Capitán, J. R. Martínez-de-Dios, and A. Ollero, "Experimental results in multi-UAV coordination for disaster management and civil security applications," *J. Intell. Robot. Syst.*, vol. 61, nos. 1–4, pp. 563–585, Jan. 2011.

[4] Y. Guo, Z. Xu, and J. Saleh, "Collaborative allocation and optimization of path planning for static and mobile sensors in hybrid sensor networks for environment monitoring and anomaly search," *Sensors*, vol. 21, no. 23, p. 7867, Nov. 2021.

[5] L. Freda et al., "3D multi-robot patrolling with a two-level coordination strategy," *Auto. Robots*, vol. 43, no. 7, pp. 1747–1779, Oct. 2019.

[6] A. Bandarupalli, D. Swarup, N. Weston, and S. Chaterji, "Persistent airborne surveillance using semi-autonomous drone swarms," in *Proc. 7th Workshop Micro Aerial Vehicle Netw., Syst., Appl.*, Jun. 2021, pp. 19–24.

[7] L. Huang, M. C. Zhou, and K. Hao, "Non-dominated immune-endocrine short feedback algorithm for multi-robot maritime patrolling," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 362–373, Jan. 2020.

[8] L. Huang, M. Zhou, K. Hao, and H. Han, "Multirobot cooperative patrolling strategy for moving objects," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 5, pp. 2995–3007, May 2023.

[9] L. Huang, M. Zhou, K. Hao, and E. Hou, "A survey of multi-robot regular and adversarial patrolling," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 4, pp. 894–903, Jul. 2019.

[10] S. C. Mohamed, S. Rajaratnam, S. T. Hong, and G. Nejat, "Person finding: An autonomous robot search method for finding multiple dynamic users in human-centered environments," *IEEE Trans. Autom. Sci. Eng. (from July 2004)*, vol. 17, no. 1, pp. 433–449, Jan. 2020.

[11] J. Roldán, P. Garcia-Aunon, M. Garzón, J. De León, J. D. Cerro, and A. Barrientos, "Heterogeneous multi-robot system for mapping environmental variables of greenhouses," *Sensors*, vol. 16, no. 7, p. 1018, Jul. 2016.

[12] F. Abbasi, A. Mesbahi, and J. M. Velni, "A new Voronoi-based blanket coverage control method for moving sensor networks," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 1, pp. 409–417, Jan. 2019.

[13] J. Yu, S. Karaman, and D. Rus, "Persistent monitoring of events with stochastic arrivals at multiple stations," *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 521–535, Jun. 2015.

[14] F. Pasqualetti, J. W. Durham, and F. Bullo, "Cooperative patrolling via weighted tours: Performance analysis and distributed algorithms," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1181–1188, Oct. 2012.

[15] A. de Luna Almeida, P. M. de Castro, T. Menezes, and G. Ramalho, "Combining idleness and distance to design heuristic agents for the patrolling task," in *Proc. Brazilian Workshop Games Digit. Entertainment*, 2003, pp. 33–40.

[16] S. Alamdari, E. Fata, and S. L. Smith, "Persistent monitoring in discrete environments: Minimizing the maximum weighted latency between observations," *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 138–154, 2014.

[17] V. Yanovski, I. A. Wagner, and A. M. Bruckstein, "A distributed ant algorithm for efficiently patrolling a network," *Algorithmica*, vol. 37, no. 3, pp. 165–186, Jul. 2003.

[18] H. Santana, G. Ramalho, V. Corruble, and B. Ratitch, "Multiagent patrolling with reinforcement learning," in *Proc. 3rd Int. Joint Conf. Auto. Agents Multiagent Syst. (AAMAS)*, vol. 4, Jul. 2004, pp. 1122–1129.

[19] A. Farinelli, L. Iocchi, and D. Nardi, "Distributed on-line dynamic task assignment for multi-robot patrolling," *Auto. Robots*, vol. 41, no. 6, pp. 1321–1345, Aug. 2017.

[20] I. A. Wagner, Y. Altshuler, V. Yanovski, and A. M. Bruckstein, "Cooperative cleaners: A study in ant robotics," *Int. J. Robot. Res.*, vol. 27, no. 1, pp. 127–151, Jan. 2008.

[21] P. A. Sampaio, R. da Silva Sousa, and A. N. Rocha, "Reducing the range of perception in multi-agent patrolling strategies," *J. Intell. Robot. Syst.*, vol. 91, no. 2, pp. 219–231, Aug. 2018.

[22] T. Wang, P. Huang, and G. Dong, "Modeling and path planning for persistent surveillance by unmanned ground vehicle," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 4, pp. 1615–1625, Oct. 2021.

[23] D. Portugal, M. S. Couceiro, and R. P. Rocha, "Applying Bayesian learning to multi-robot patrol," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Oct. 2013, pp. 1–6, doi: 10.1109/SSRR.2013.6719325.

[24] X. Zhou, W. Wang, T. Wang, Y. Lei, and F. Zhong, "Bayesian reinforcement learning for multi-robot decentralized patrolling in uncertain environments," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 11691–11703, Dec. 2019.

[25] T. Menezes, P. C. A. R. Tedesco, and G. Ramalho, "Negotiator agents for the patrolling task," in *Advances in Artificial Intelligence—IBERAMIA-SBIA 2006*, vol. 4140. Berlin, Germany: Springer, Oct. 2006, doi: 10.1007/11874850_9.

[26] A. Machado, G. Ramalho, J.-D. Zucker, and A. Drogoul, "Multiagent patrolling: An empirical analysis of alternative architectures," in *Multi-Agent-Based Simulation II*, vol. 2581. Berlin, Germany: Springer, Apr. 2003, pp. 155–170.

[27] Y. Chevaleyre, "Theoretical analysis of the multi-agent patrolling problem," in *Proc. IEEE/WIC/ACM Int. Conf. Intell. Agent Technol.*, Sep. 2004, pp. 302–308, doi: 10.1109/IAT.2004.1342959.

[28] D. Portugal and R. Rocha, "MSP algorithm: Multi-robot patrolling based on territory allocation using balanced graph partitioning," in *Proc. ACM Symp. Appl. Comput.*, Mar. 2010, pp. 1271–1276, doi: 10.1145/1774088.1774360.

[29] D. Shen et al., "Test scenarios development and data collection methods for the evaluation of vehicle road departure prevention systems," *IEEE Trans. Intell. Vehicles*, vol. 4, no. 3, pp. 337–352, Sep. 2019.

[30] J. Yin, D. Shen, X. Du, and L. Li, "Distributed stochastic model predictive control with Taguchi's robustness for vehicle platooning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15967–15979, Sep. 2022.

[31] D. Shen, Y. Chen, and L. Li, "State-feedback switching linear parameter varying control for vehicle path following under uncertainty and external disturbances," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 3125–3132, doi: 10.1109/ITSC55140.2022.9922086.

[32] Y. Chen and L. Li, *Advances in Intelligent Vehicles*. New York, NY, USA: Academic, 2014, pp. 291–318.

[33] T. Wang, P. Huang, and G. Dong, "Cooperative persistent surveillance on a road network by multi-UGVs with detection ability," *IEEE Trans. Ind. Electron.*, vol. 69, no. 11, pp. 11468–11478, Nov. 2022.

[34] T. Wang, G. Dong, and P. Huang, "Pheromone-diffusion-based conscientious reactive path planning for road network persistent surveillance," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 7922–7928, doi: 10.1109/ICRA48506.2021.9561707.

[35] E. Hernandez, A. Barrientos, and J. del Cerro, "Selective smooth fictitious play: An approach based on game theory for patrolling infrastructures with a multi-robot system," *Expert Syst. Appl.*, vol. 41, pp. 2897–2913, May 2014.

[36] E. Sless, N. Agmon, and S. Kraus, "Multi-robot adversarial patrolling: Facing coordinated attacks," in *Proc. Int. Conf. Auto. Agents Multi-Agent Syst.*, May 2014, pp. 1093–1100.

[37] N. Agmon, S. Kraus, and G. A. Kaminka, "Multi-robot perimeter patrol in adversarial settings," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 2339–2345, doi: 10.1109/ROBOT.2008.4543563.

[38] T. Sak, J. Wainer, and S. Goldenstein, "Probabilistic multiagent patrolling," in *Advances in Artificial Intelligence—SBIA 2008*, vol. 5249. Berlin, Germany: Springer, Oct. 2008, pp. 124–133.

[39] T. Alam, "Decentralized and nondeterministic multi-robot area patrolling in adversarial environments," *Int. J. Comput. Appl.*, vol. 156, no. 2, pp. 1–8, Dec. 2016.

[40] L. Xu, "Learning and planning under uncertainty for green security," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 4927–4928.

[41] T. Alam, M. M. Rahman, L. Bobadilla, and B. Rapp, "Multi-vehicle patrolling with limited visibility and communication constraints," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2017, pp. 465–470.

[42] N. Agmon, G. A. Kaminka, and S. Kraus, "Multi-robot adversarial patrolling: Facing a full-knowledge opponent," *J. Artif. Intell. Res.*, vol. 42, pp. 887–916, Dec. 2011.

[43] N. Agmon, "On events in multi-robot patrol in adversarial environments," in *Proc. Int. Conf. Adapt. Agents Multi-Agent Syst.*, vol. 1, Jan. 2010, pp. 591–598.

[44] E. Sless Lin, N. Agmon, and S. Kraus, "Multi-robot adversarial patrolling: Handling sequential attacks," *Artif. Intell.*, vol. 274, pp. 1–25, Sep. 2019.

[45] M. Chen, Z. Zhou, and C. J. Tomlin, "A path defense approach to the multiplayer reach-avoid game," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 2420–2426, doi: 10.1109/CDC.2014.7039758.

[46] R. Yan, Z. Shi, and Y. Zhong, "Task assignment for multiplayer reach-avoid games in convex domains via analytical barriers," *IEEE Trans. Robot.*, vol. 36, no. 1, pp. 107–124, Feb. 2020.

[47] S. L. Smith, S. D. Bopardikar, and F. Bullo, "A dynamic boundary guarding problem with translating targets," in *Proc. 48th IEEE Conf. Decis. Control (CDC) 28th Chin. Control Conf.*, Dec. 2009, pp. 8543–8548.

[48] S. Bajaj and S. D. Bopardikar, "Dynamic boundary guarding against radially incoming targets," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Dec. 2019, pp. 4804–4809.

[49] D. G. Macharet, A. K. Chen, D. Shishika, G. J. Pappas, and V. Kumar, "Adaptive partitioning for coordinated multi-agent perimeter defense," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 7971–7977.

[50] D. A. Anisi, P. Ögren, and X. Hu, "Cooperative minimum time surveillance with multiple ground vehicles," *IEEE Trans. Autom. Control*, vol. 55, no. 12, pp. 2679–2691, Dec. 2010.

[51] P. Wu, F. Chu, A. Che, and M. Zhou, "Bi-objective scheduling of fire engines for fighting forest fires: New optimization approaches," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1140–1151, Apr. 2018.

[52] F. Saidi, Z. Trabelsi, and H. B. Ghezala, "Towards a multidimensional model for terrorist attacks analysis and mining," in *Proc. 28th Int. Conf. Comput. Theory Appl. (ICCTA)*, Oct. 2018, pp. 55–59, doi: 10.1109/ICCTA45985.2018.9499167.

[53] M. A. Berawi et al., "Optimizing search and rescue personnel allocation in disaster emergency response using fuzzy logic," *Int. J. Technol.*, vol. 13, no. 7, pp. 1416–1426, Nov. 2019.

[54] X. Lin and C. G. Cassandras, "An optimal control approach to the multi-agent persistent monitoring problem in two-dimensional spaces," *IEEE Trans. Autom. Control*, vol. 60, no. 6, pp. 1659–1664, Jun. 2015.

[55] W. S. Cheong, S. F. Kamarulzaman, and M. A. Rahman, "Implementation of robot operating system in smart garbage bin robot with obstacle avoidance system," in *Proc. Emerg. Technol. Comput., Commun. Electron. (ETCCE)*, Dec. 2020, pp. 1–6, doi: 10.1109/ETCCE51779.2020.9350912.

[56] L.-B. Chen, X.-R. Huang, W.-H. Chen, W.-Y. Pai, G.-Z. Huang, and W.-C. Wang, "Design and implementation of an artificial intelligence of things-based autonomous mobile robot system for cleaning garbage," *IEEE Sensors J.*, vol. 23, no. 8, pp. 8909–8922, Apr. 2023.

[57] D. Portugal and R. P. Rocha, "Multi-robot patrolling algorithms: Examining performance and scalability," *Adv. Robot.*, vol. 27, no. 5, pp. 325–336, Apr. 2013.

[58] R. W. Floyd, "Algorithm 97: Shortest path," *Commun. ACM*, vol. 5, no. 6, p. 345, Jun. 1962.

[59] M. Cui, L. Li, M. Zhou, J. Li, A. Abusorrah, and K. Sedraoui, "A bi-population cooperative optimization algorithm assisted by an autoencoder for medium-scale expensive problems," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 11, pp. 1952–1966, Nov. 2022.

[60] M. Cui, L. Li, M. Zhou, and A. Abusorrah, "Surrogate-assisted autoencoder-embedded evolutionary optimization algorithm to solve high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 26, no. 4, pp. 676–689, Aug. 2022.

[61] C. Lin, Z. Cao, and M. Zhou, "Autoencoder-embedded iterated local search for energy-minimized task schedules of human–cyber–physical systems," *IEEE Trans. Autom. Sci. Eng.*, early access, May 4, 2023, doi: 10.1109/TASE.2023.3267714.

[62] H. Yuan, Q. Hu, J. Bi, J. Lü, J. Zhang, and M. Zhou, "Profit-optimized computation offloading with autoencoder-assisted evolution in large-scale mobile-edge computing," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11896–11909, Feb. 2023, doi: 10.1109/JIOT.2023.3244665.

[63] J. Yin et al., "Reliable global path planning of off-road autonomous ground vehicles under uncertain terrain conditions," *IEEE Trans. Intell. Vehicles*, vol. 9, no. 1, pp. 1161–1174, Jan. 2024.

[64] D. Cao et al., "Future directions of intelligent vehicles: Potentials, possibilities, and perspectives," *IEEE Trans. Intell. Veh.*, vol. 7, no. 1, pp. 7–10, Mar. 2022.

[65] X. Wang, S. Han, L. Yang, T. Yao, and L. Li, "Parallel Internet of Vehicles: ACP-based system architecture and behavioral modeling," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3735–3746, May 2020.

JunQi Zhang (Senior Member, IEEE) received the Ph.D. degree in computing science from Fudan University, Shanghai, China, in 2007. He was a Post-Doctoral Research Fellow with Peking University, Beijing, China, in 2007. From 2014 to 2015, he was a Visiting Scholar with New Jersey Institute of Technology, Newark, NJ, USA. He is currently a Full Professor with Tongji University, Shanghai. He has published more than 20 articles in IEEE TRANSACTIONS and more than 60 papers in conferences. His current research interests include swarm intelligence, swarm robots, multi-agent systems, learning automata, reinforcement learning, and big data. He was a recipient of the Outstanding Postdoctoral Award with Peking University in 2007.

YunZhe Wu received the B.S. degree in computer science and technology from Dalian Maritime University, China, in 2022. He is currently pursuing the M.S. degree with the Department of Computer Science Technology, Tongji University, Shanghai, China. His current research interests include collaborative mechanisms between multiple intelligences, robotic cooperative surveillance of the environment, and localization and processing of emergent targets.

MengChu Zhou (Fellow, IEEE) received the Ph.D. degree from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990. He joined New Jersey Institute of Technology, where he is currently a Distinguished Professor. He has over 1200 publications, including 17 books, more than 850 journal articles (more than 650 in IEEE TRANSACTIONS), 31 patents, and 32 book-chapters. His research interests include Petri nets, automation, the Internet of Things, and big data. He is a fellow of IFAC, AAAS, CAA, and NAI.