IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

IEEE VEHICULAR TECHNOLOGY SOCIETY SECTION

Received 6 September 2022, accepted 2 October 2022, date of publication 5 October 2022, date of current version 12 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3212079

RESEARCH ARTICLE

A Local PSO-Based Algorithm for Cooperative Multi-UAV Pollution Source Localization

HASSAN SAADAOUI AND FAISSAL EL BOUANANI, (Senior Member, IEEE)
ENSIAS College of Engineering, Mohammed V University, Rabat 10000, Morocco
Corresponding author: Faissal El Bouanani (f.elbouanani@um5s.net.ma)

ABSTRACT Recently, air pollution has grown significantly, and it can frequently be challenging to identify the sources of contaminants. This article studies the deployment of multi-cooperative unmanned aerial vehicles (UAVs) to look for sources of pollution in an unknown region. Specifically, a probabilistic search strategy based on the Local Particle Swarm Optimization (*LoPSO*) method is suggested to design an optimal strategy that enhances the cooperative search of drones while cutting down on overall search time and improving source detection efficiency. The entire strategy is divided into two phases: exploration and exploitation. In the first phase, the detection and tracing of the plume is the favored task, in which each UAV operates in either the *Greedy* or *LoPSO* mode and chooses its path based on plane coordinates generated according to the active mode. By utilizing the areas with a high probability of discovering the source on flight mode *LoPSO*, the search is focused during the exploitation phase on the precise search of the exact location of the pollutant source. This is done under the direction of a probabilistic computation that uses the Bayesian process model to create and update the probability map of the pollutant source location as new sensor data becomes available. The simulation data of the proposed technique demonstrates promising results in terms of the complexity and accuracy attained in identifying pollution sources.

INDEX TERMS Cooperative search, decision making, greedy, particle swarm optimization, pollution source, unmanned aerial vehicle.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) employment in the military and civilian sectors has risen in importance as modern science and technology have evolved. In particular, there are numerous areas where they may be beneficial for completing complicated or dangerous tasks [1], [2]. Equipped with onboard sensors, they can be used for environmental monitoring, including smokestacks, land and industrial vehicles, search and rescue operations for natural or man-made disasters, and exploration of urban or natural areas. In recent years, researchers have been drawn to drones because of their simplicity, adaptability, and agility. However, if future tasks become more complex, a single UAV system will have its own limitations and may not be able to meet the demands of many areas. Due to their great self-organization and their scalability, swarm drone systems have aroused interest, especially the control cooperation, which has become one of the most interesting themes and the most important topics in the UAV applications field.

Research work in the area of multi-UAV cooperation is of the utmost importance, especially in the area of air pollution monitoring, which is one of the most serious threats to human health, representing one in nine deaths worldwide [3]. Some of the current and long-term health problems are associated with ambient air pollution include breathing issues, cardiovascular illness, and lung cancer [4], [5]. which has led to the development of tools and technology that enable environmental monitoring. In this context, UAVs appear to be one of the most revolutionary solutions to address the pollution issue in this area.

Indeed, air pollution monitoring applications using unmanned aerial vehicles (UAVs) [6], [7] demonstrate significant advantages over traditional approaches such as field stations or satellite imagery: high precision, low cost, security, adaptability, multi-function, and ease of implementation [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Jie Gao.

Thus, when it comes to air pollution monitoring, locating the source of the pollution in a timely and efficient manner is critical. This is especially can be crucial in situations when the fast detection of these sources might avert severe catastrophes, like the release of gas substances as a result of industrial disasters. Tracking and monitoring such changing polluted areas is difficult using traditional monitoring approaches. Multiple intelligent monitoring UAVs in autonomous sensing provide novel monitoring options. The challenge of pollution source detection using UAVs is divided into two phases [9], namely (i) location of the air pollution plume, and (ii) pollutant plume tracking to the source.

A. RELATED WORK

The use of robotic technology to locate polluting sources has advanced quickly in recent years. The proposed algorithms may be generally classified into three groups based on the various studies: (i) an engineering strategy algorithm, (ii) bio-inspired algorithm, and (iii) swarm intelligence optimization algorithm based on multi-agent reinforcement learning.

For instance, the authors of [10] and [11] sample the search region with various mobile robots. The movement of each robot is controlled by global optimization algorithms. In particular, [10] describes a leading-follower approach for guiding mobile robots utilizing a particle swarm optimization (PSO) algorithm. Each robot is viewed as a swarm particle, and the movement is guided by the Schrödinger equation. The swarm's leader is determined by the global ideal position. By measuring and navigating in the leader's favored direction, the followers help the leader. The research studies in [12] and [13] concentrate on minimizing the time it takes to complete a task and information sharing among individual robotic agents.

In [14], two experiments were developed: one that takes airflow information into account, and the other that does not. With displacement or mixing ventilation, the airflow is altered. A swarm of six terrestrial robots is steered through three steps for each experiment: detecting the plume using a random divergence technique, monitoring the plume using a basic and an improved version of the whale optimization algorithm (WOA), and locating the pollutant source.

Recent literature has described some studies that employed UAVs to identify pollution sources. Reference [15] proposes an algorithm for pollutant concentration monitoring in particular places using the meta-heuristic chemotaxis approach and particle swarm intelligence (SI). To undertake a coordinated scan of the search area, the four UAV agents used in [12] share their location, velocity, and formation vector. The following steps are carried out during the exploratory phase: a circular formation is created around the leader, the swarm is moving in a logarithmic spiral, and the UAV that reacts to a gas measurement becomes the leader. If the ith UAV detects a greater gas concentration than previous readings, that UAV is selected as the new leader. The studies start with the assumption that gas concentration falls as one gets away from the source and simulate the plume using a Gaussian model. Three methods are applied during the exploration phase: Brownian motion behavior, random walk scanning with possible drone orientations, and leader-follower. The authors of [16] provide concepts for UAV formation using a leader-follower technique, as well as controls for monitoring industrial pollutants, including tracking oil spills and pollutant particle clouds.

A research presented in [17] presents a system that uses numerous UAVs to monitor predetermined waypoints to detect nuclear radiation levels. The detection of radioactive radiation was tested utilizing a cooperative technique including two UAVs. Using several coordinated quadrotors, an approach for source localization is examined in [18]. The cyber and physical components of this technique are the quadrotor section and the control law section, respectively. The developed controller uses the flow velocity, gradient, and divergence of the pollutant concentration at the UAVs' landing point to determine the direction that the pollution plume will counter-propagate.

Moreover, due to its efficiency and simplicity, Gradient descendent (GD)-based method has been widely employed as a deterministic trainer. In [19] it was combined with a self-tuning grey wolf optimizer (STGWO) to improve the learning ability of the mixed multilayer wavelet neural network (MMWNN). Many other gradient-based strategies for determining a pollutant's concentration are ''bio-inspired'', in that they are based on the pheromone tracking or foraging behavior of bacteria or insects [20], [21], [22].

Although the aforementioned algorithms have enabled multi-UAV cooperative search. Likewise, they would still face a number of difficulties, including a lack of an effective cooperation structure, good energy management, and most importantly, a lack of an effective process for the exchange of environmental information. To remedy the issue, SI-based algorithms have been rolled out. However, SI a subset of artificial intelligence, has been widely employed in the literature due to its ability to generate excellent and computationally tractable solutions while assuring convergence and robustness [23]. Moreover, the SI makes it possible to investigate the complex and collective behavior of systems made up of a large number of tiny components that are capable of local and remote communication.

Moreover, it has been included in a variety of processes to tackle difficult real-world issues in both science and industry. In [24], for example, a distributed intelligent resource scheduling system is presented, allowing each agent launched on each mobile edge computing server to carry out both centralized training and decentralized execution. Another use of SI is described in [25], in which the PSO is used to solve a resource scheduling problem in UAV-MEC systems and offers training data for deep neural networks. Accordingly, cooperation between the UAVs is guaranteed. Yet, there is still a concern about how well these methods control their overall energy consumption. In order to accomplish a search mission in an uncharted area, a swarm of autonomous drones can be automatically and adaptively coordinated by putting in place a construction mechanism for

sub-swarm [26], [27]. This strategy enables independent local searches while optimizing the usage of the swarm's overall energy.

B. CONTRIBUTION

This study presents the outcomes of using an intelligent technique for detecting an air pollution source in the absence of favorable environmental conditions. Motivated by the above, we intend to build cooperative, autonomously piloted drones that can quickly detect pollution sources. Some considerations were made for simplicity to create a controlled environment that was comparable enough to real-life events. For this purpose, it is believed that once the chemical pollutant has been detected, the UAV sensors will recognize it. Moreover, this study does not include image processing techniques, motion control, limited actuator control, or other types of sensors.

We present a meta-heuristic algorithm titled local PSO (*LoPSO*) [27] for enhanced cooperative pollutant source finding by dividing the whole swarm into sub-swarms based on communication range and combining a gradient-based approach with a probability-based technique.

The drone swarm system, like the intelligent swarm system, is made up of member drones that work together through multi-sensor information fusion. It is hypothesized that a pollution source can continually create chemical filaments in an unknown complicated environment. When a UAV discovers a pollutant, it executes a self-organized task assignment based on the degree of pollutant detection and announces its status information to enter the coordinated search state; if the pollutant is not detected, the member UAV remains in the roaming state.

The work detailed in reference [28] served as the foundation for the probabilistic component, which use the Bayesian approach to create and update a probability map of the location of the polluting source when new sensor data becomes available while the drone is flying in the region. To evaluate the suggested strategy, the polluted area was re-created using a particle distribution model with turbulence for wind influences. More specifically, the following summarizes our key contribution:

- We propose a hybrid technique that relies on training sub-swarms to communicate various useful information based on LoPSO with the divide and conquer strategy using local communication networks.
- We provide an enhancement of the PSO parameters to enable a quick convergence in order to balance local and global exploration by the sub-swarm.
- We present a hybrid strategy that combines decentralized and centralized designs, which often presuppose the formation of sub-swarms and reduces communication flows.
- We demonstrate that our proposed approach outperforms the well-known robust algorithm in order to better understand performance and computational complexity.

C. ORGANIZATION OF THE PAPER

The remaining contents of this article are arranged as follows. The model system is described in Section II, along with the mathematical model of the algorithm, the simulation environment, the air pollution distribution model, and the UAV model that was employed. The proposed search method to address the localization of pollution sources is presented in Section III. We conduct the simulations experiments in section IV to validate the suggested search strategy and set up a comparison analysis. Finally, in section V, conclusions are drawn.

II. SYSTEM MODEL

As seen in Fig. 1, a rectangular geographical area Ω is divided into $N_x \times N_y$ equitable rectangular cells of length h_c, defined by $\Omega = \{C_1, \ldots, C_M\}$, where $M = N_x \times N_y$, with N_x and N_y denoting the number of rows and columns, respectively. Therein, N_u UAVs $\{U_k\}_{1 \leqslant k \leqslant N_u}$ are collaboratively deployed to detect an air pollution source that is considered to be static and is located in a location that is continuously generating a polluting gas. The objective is to locate the source, which is thought to be where the gas concentration is highest. As one gets further away from the source, the gas concentration often drops radially.

Furthermore, the UAVs are supposed to proceed unaffected by wind speeds and fly over the considered area at slightly different heights to avoid possible collisions. In the following, the cell C_i is referenced by its center for brevity, i.e, $\{C_i\}_{1 \leqslant i \leqslant M} \triangleq (x_i, y_i)$.

Let's begin by introducing the following functions for the sake of clarity:

- $B_k(n, I)$: broadcasting the data I outlined in Table 2, by either BS if $k = 0$ or U_k when $k \geq 1$, with n defines the transmitted data type and I the possible values that are shown in Table 1.
- $R_k(n, I)$: receiving the data I of type n from either BS if $k = 0$ or U_k when $k \geq 1$. Noted that if a broadcast is detected, such a function is called.
- $GetPolVal(i, t)$: getting pollution values above a cell C_i. It returns the value detected by the pollution sensor in this cell at a given time t.

A. UAV DYNAMIC MODEL

The UAVs are modeled as point masses moving synchronously in discrete time with minimal turning radius. They are also assumed to be symmetrical with the same transmit power and altitude with limited sensor range. They move unaffected by wind speeds. Moreover, U_k takes decisions by updating its current location at each $t = p\tau$ with $p \in \mathbb{N}$, and either goes to another cell, depending on its own flying information and that of its neighbors or remains indifferent. Furthermore, without being aware of the source's location, vehicles can measure the emission intensity, the decay profile, or the gas concentration produced by the source at their current positions. The purpose is to pinpoint the source, which is expected to be static and located in a large region

TABLE 1. List of symbols.

Symbol	*Description*	*Symbol*	*Description*
N_u	Number of UAVs	U_k	kth UAV
C_i	ith cell$i$	C_s	Pollutant source cell
$\psi_i^{(k)}(t)$	Pollutant's particle density at time t read by U_k located above C_i	μ_p	Pollutant detection confirmation threshold
μ_s	Difference threshold between two detections of the pollutant	μ_o	Threshold for sigmoid function
$\mathcal{V}^{(k)}(t)$	U_k's velocity	d_r	UAV's range of communication
$\left\lvert\mathcal{N}_\varphi^{(k)}(t)\right\rvert$	$\mathcal{N}_\varphi^{(k)}(t)$'s elements number	h_c	cell length
$\mathcal{N}_l^{(k)}(t)$	List of indices identifying U_k's long neighbours within d_r at a given time t	$\mathcal{N}_s^{(k)}(t)$	List of indices identifying U_k's temporary neighbours within d_r at a given time t
N_c	Number of cells covered by the UAV	$D_{\max}$	Maximum time for a search
t_{PSO}	PSO maximum time	τ	Sensing period
E_t	Minimum Energy level at which to Stop searching	E_k	U_k's energy
F_k	U_k's flight mode	H_k	U_k's flight height
$\chi^{(k)}(t)$	U_k's position at t	M	the number of cells in the map
$\delta^{(k)}(t)$	U_k's best position at time t	$\Delta^{(k)}(t)$	Best position of the U_k's sub-swarm at time t
$Conf$	Average confirmation time of the detected source's location	$g_{\max}$	maximum fitness function
$P_{\max}$	maximum validation probability of source location,	ST	Average confirmation time of the detected source's location
ω_{set}	Particle settling velocity	W_{dep}	Deposition coefficient
Q	Pollutant release rate	H	Center-line effective height of the plume
σ_υ	Gaussian distributions' standard deviation, showing how much the plume has expanded in the υ direction, $\upsilon = x, y,$ or z	$\lambda^{(k)}(t)$	Inertia weight value
c_1	UAV's Individual cognitive coefficients	c_2	UAV's Social cognitive coefficients
r_1 and r_2	Random values referring to the acceleration coefficients		

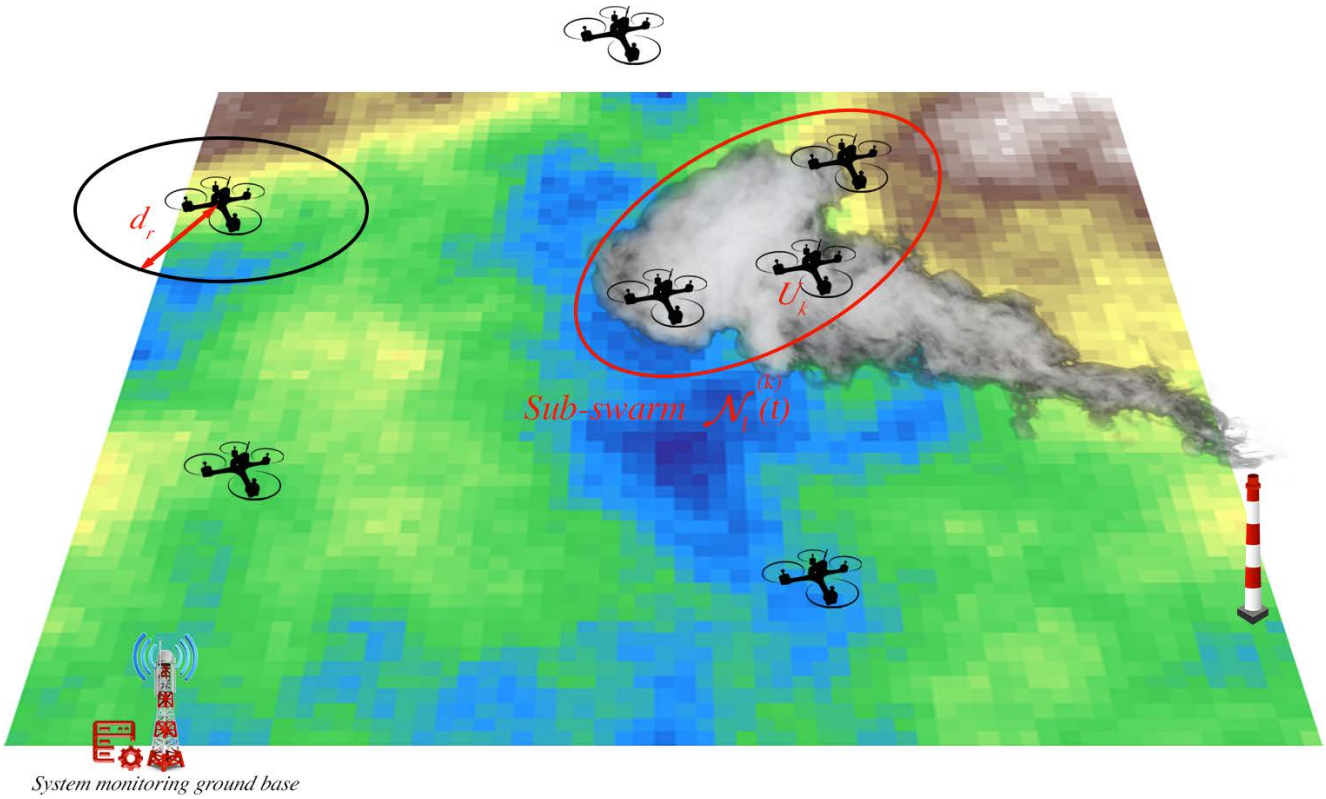

FIGURE 1. Illustrative example of cooperative UAVs tracking plume air pollutant.

continually spewing polluting gas where its concentration is highest.

The polluting gas concentration usually decreases radially as the distance from the source rises. Unfortunately,

TABLE 2. Possible values of the data *I*.

n	*Data I to receive / broadcast*
0	True : confirmation for joining a sub-swarm False: message from BS ordering the search to end Who : message from a UAV to search for nearby UAVs Free : message to leave a sub-swarm Join : an invitation to join a sub-swarm
1	$P^{(k)}(t)$: U_k's probability map
2	$\Psi^{(k)}(t)$: U_k's plume density map
3	$\mathcal{N}_{\varphi}^{(k)}(t)$: set of indices referring to the neighbors of U_k
4	χ_s: pollutant source position
5	$\delta^{(k)}(t)$: U_k's best position at time t $\Delta^{(k)}(t)$: the best position of the U_k's sub-swarm at time t

describing the immediate shape of a polluting plume in a turbulent flow is extremely difficult (if not impossible). Alternatively, we may utilize the information acquired from the UAVs to predict the probable locations of the sources by creating probabilistic models of the spatial and temporal development of the pollutant plume.

For the purpose of simplicity, we assume that U_k has two main modes and flies in a certain direction [27]:

i) *Greedy* mode, wherein the UAV flies toward cells with the highest pollutant concentration value at speed $\mathcal{V}_{\text{avg}}^{(k)}(t)$;
ii) the *LoPSO* mode, in which the UAV moves inside a sub-swarm to confirm, cooperatively, the location of a pollution source in a reduced search duration at speed $\mathcal{V}_{\min}^{(k)}(t) \leq \mathcal{V}^{(k)}(t) \leq \mathcal{V}_{\max}^{(k)}(t)$.

B. POLLUTANT DISTRIBUTION MODEL

The dispersion of air pollution can be considered as a level of pollutant concentration in the atmosphere that fluctuates over time, where the source makes up the maximum of the dispersion function. By accounting for two phenomena, the diffusion of pollutants in the atmosphere and the movement of these dispersed concentrations owing to the wind, an optimization technique may be adopted to follow the pollution plume and determine the maximum of the dispersion function [29], [30].

The pollutant distribution model utilized throughout this is based on the particle diffusion method introduced in [30], which is based on two assumptions: particle number conservation, and the relationship between flux and density. Let $\psi(\vec{\chi}, t)$ $\left[\text{kg/m}^3\right]$ denotes the pollutant's particle density at time t and position $\vec{\chi}$. The law of mass conservation for $\psi(\vec{\chi}, t)$ can be represented as [31]

$$\frac{\partial \psi(\vec{\chi}, t)}{\partial t} + \nabla \vec{\Gamma}(\vec{\chi}, t) = f_s(\vec{\chi}, t), \tag{1}$$

where $\vec{\Gamma}(\vec{\chi}, t)$ denotes the pollutant flux due to advection and diffusion, and $f_s(\vec{\chi}, t)$ $\left[\text{kg/m}^3\text{s}\right]$ refers to the source function.

The diffusive contribution to the flux is caused by turbulent eddy motion in the atmosphere and is generally assumed to follow Fick's law [32]

$$\vec{\Gamma}_D = -D\nabla\psi(\vec{\chi}, t)), \tag{2}$$

where $D = (D_x, D_y, D_z)$ is a diagonal matrix with the eddy diffusion coefficients usually being functions of the position. The advection is the effect of wind components, which advect pollutants downstream in a simple linear flow, is described by a linear equation [32]

$$\vec{\Gamma}_A = \psi(\vec{\chi}, t)\vec{\omega}(t), \tag{3}$$

where $\vec{\omega}(t) = (\omega_x(t), \omega_y(t), \omega_z(t))$ is the wind velocity vector. Adding together the two fluxes and setting $\vec{\Gamma} = \vec{\Gamma}_A + \vec{\Gamma}_D$, the pollutant model can be expressed as

$$\frac{\partial \psi(\vec{\chi}, t)}{\partial t} + \nabla.(\psi(\vec{\chi}, t)\vec{\omega}(t)) = \nabla(D\nabla\psi(\vec{\chi}, t)) + s(\vec{\chi}, t). \tag{4}$$

Many boundary conditions can be specified. Nonetheless, we are interested in solving this equation in the half-space $z \geq 0$ where $z = 0$ refers to the ground. We also limit ourselves to short-range pollutant dispersion, such as when the concentration tends to zero in a distant area. This is in line with the domain's contaminant mass conservation. The ground-level boundary condition is more relevant since this is where contaminants are deposited. That is,

$$\omega_z\psi(\vec{\chi}, t) - D_z\frac{\partial \psi(\vec{\chi}, t)}{\partial z} = 0 \text{ at } z = 0. \tag{5}$$

If we set $\omega_z = -\omega_{\text{set}}$ as the particle settling velocity and W_{dep} as the deposition coefficient, reflecting the impact of the overall flow of contaminants penetrating the ground, the Robbin boundary condition is obtained

$$-\omega_{\text{set}}\psi(\vec{\chi}, t) - D_z\frac{\partial \psi(\vec{\chi}, t)}{\partial z} = -W_{\text{dep}}\psi(\vec{\chi}, t), \ z \neq 0. \tag{6}$$

The solution to this equation is given in [30] as

$$\psi(x, y, z) = \frac{Q}{2w\pi\sigma_y\sigma_z}\exp\left(\frac{-y^2}{2\sigma_y^2}\right)\left[\exp\left(\frac{-(z-H)^2}{2\sigma_z^2}\right) + \exp\left(\frac{-(z+H)^2}{2\sigma_z^2}\right)\right], \ z \neq 0, \tag{7}$$

where Q represents the pollutant release rate, w as the wind speed, H represents the center-line effective height of the plume, and σ_y and σ_y refer to the standard deviation of the Gaussian concentration distribution.

C. SOURCE PROBABILITY MAP MODEL

During the UAV navigation, the search is performed using a finite probability map of the search area [28]. Therein, meshed indicates the probability of locating the pollution source in a certain place. This method is predicated on the drone's ability to measure or infer the wind speed and pollutant concentration at its current location.

Let $X^{(k)}(t)_{i,j}$ represent the event that the source is located in the cell C_i at time t when the U_k, which is in another cell C_j, detects or does not the pollutant at t. Additionally, a series of pollutant detection or non-detection will take place

along the UAV path while following a plume. At a given instant t, U_k through a chemical detector characterized by imperfect detection accuracy and communication capabilities, scans the cell C_j the observation obtained is binary: (i) $Z_j^{(k)}(t) = 1$ indicating the detection of the pollutant in cell C_j, or (ii) $Z_j^{(k)}(t) = 0$ indicating that U_k had not detected any pollutants when flying over this cell at that time.

Consequently, the creation of the map using the Bayesian process is described as:

$$P^{(k)}(t) = \left\{ P_{i,j}^{(k)}(t), j = 1..M \right\}, \tag{8}$$

where

$$P_{i,j}^{(k)}(t) = \Pr\left(X_{i,j}^{(k)}(t) | Z_j^{(k)}(t) \right). \tag{9}$$

The immediate structure of a plume in a turbulent flow is unfortunately exceedingly challenging, if not impossible, to define. Alternately, we may utilize the data gathered by the UAV to determine potential source locations by creating probabilistic descriptions of the spatial and temporal development of the pollutant plume.

The same concept utilized in [28] served as the basis for the plume simulation model that was used for the evaluation of the algorithms in this paper, whose chemical filament location is described as

$$\dot{\chi}(t) = \Gamma(\chi(t)) + \zeta(t), \tag{10}$$

where $\chi(t) = (x(t), y(t))$ denotes the chemical filament location, Γ as the mean flow velocity, and $\zeta(t)$ as a random process assumed to be Gaussian with zero mean and $[\sigma_x^2, \sigma_y^2]$ variance. Further, σ_x^2 and σ_y^2 represent the variances from a Gaussian distribution, which indicate the spread of the plume in the x and y directions, respectively. Thus,

We can obtain

$$\chi(t) = \int_t \Gamma(\chi(\theta))d\theta + \int_t \zeta(\theta)d\theta + \chi_s, \tag{11}$$

with χ_s denotes the source location.

let's denote

$$\kappa(t) = \int_t \zeta(\theta)d\theta, \tag{12}$$

and

$$\mathcal{W}(t) = \int_t \Gamma(\chi(\theta))d\theta, \tag{13}$$

which $\kappa(t) = (\kappa_x(t), \kappa_y(t))$ is a Gaussian noise process with zero mean and $[t\sigma_x^2, t\sigma_y^2]$ variance, and $\mathcal{W}(t) = (w_x(t), w_y(t))$. Substituting (12) and (13) into (11), yields

$$\chi(t) = \mathcal{W}(t) + \kappa(t) + \chi_s. \tag{14}$$

The probability density function of $\kappa(t)$ is given as [28]

$$K_x(\kappa(t)) = \frac{\exp\left(-\frac{\kappa_x^2(t)}{2t\sigma_x^2}\right)}{\sqrt{2\pi t\sigma_x^2}}, \tag{15}$$

and

$$K_y(\kappa(t)) = \frac{\exp\left(-\frac{\kappa_y^2(t)}{2t\sigma_y^2}\right)}{\sqrt{2\pi t\sigma_y^2}}, \tag{16}$$

while $\kappa_\epsilon(t) = \epsilon(t) - \epsilon_s(t) - w_\epsilon(t)$, where $\epsilon = x, y$.

Let $S_{i,j}^{(k)}(t)$ stand for the probability that a source in C_i is releasing one chemical filament at time t given that the chemical is in C_j detected by U_k

$$\begin{aligned} S_{i,j}^{(k)}(t) &= \int_{x\in C_i} K_x(\kappa(t))dx \times \int_{y\in C_i} K_y(\kappa(t))dy \\ &= \frac{1}{2\pi t\sigma_x\sigma_y} \\ &\quad \times \int_{x\in C_i} \exp\left(-\frac{(x - x_i - w_x)^2}{2t\sigma_x^2}\right)dx \\ &\quad \times \int_{y\in C_i} \exp\left(-\frac{(y - y_i - w_x)^2}{2t\sigma_y^2}\right)dy. \end{aligned} \tag{17}$$

This integral can be approximated by multiplying the center point value by the cell area due to the small cell size. Consequently, it will be approximately given as

$$\begin{aligned} S_{i,j}^{(k)}(t) \approx \frac{M}{2\pi t\sigma_x\sigma_y}\Bigg[&\exp\left(-\frac{(x_j - x_i - \omega_x(t))^2}{2t\sigma_x^2}\right) \\ &\times \exp\left(-\frac{(y_j - y_i - \omega_y(t))^2}{2t\sigma_y^2}\right)\Bigg], \end{aligned} \tag{18}$$

while M is the number of cells.

Let's define ρ_j is the probability of detecting a pollutant in cell C_j when there is a detectable pollutant in such cell. Therefore, the probability of detecting chemical released from a source in C_i at time t in cell C_j by U_k is $\rho_j S_{i,j}^{(k)}(t)$, and the probability of not detecting this chemical is $1 - \rho_j S_{i,j}^{(k)}(t)$. Subsequently, let $\beta_{i,j}^{(k)}(t)$ indicates the probability that a source will be present in cell C_i at time $t-\tau$ given the presence of detectable chemical in cell C_j at time t, assuming that the source continually released chemical commencing. Since it is uncertain that the chemical is in cell C_j, therefore

$$\beta_{i,j}^{(k)}(t) = \frac{1}{2}\left(S_{i,j}^{(k)}(t-\tau) + S_{i,j}^{(k)}(t)\right). \tag{19}$$

the probability of not detecting the chemical in cell C_j at time t due to the continuous release from a source in cell C_i is

$$\gamma_{i,j}^{(k)}(t) = \left(1 - \rho S_{i,j}^{(k)}(t-\tau)\right)\left(1 - \rho S_{i,j}^{(k)}(t)\right). \tag{20}$$

Each UAV has a different probability of identifying a pollutant in a specific cell during every time step. The Bayesian rule [28] is used to recalculate these probabilities following each visit of the cells between $t - \tau$ and t as

$$P_{i,j}^{(k)}(t) = \begin{cases} MP_{i,j}^{(k)}(t-\tau)\beta_{i,j}^{(k)}(t), & Z_j^{(k)}(t) = 1 \\ M\frac{P_{i,j}^{(k)}(t-\tau)\gamma_{i,j}^{(k)}(t)}{\sum_{i=1}^{M}\gamma_{i,j}^{(k)}(t)}, & Z_j^{(k)}(t) = 0 \end{cases}, \tag{21}$$

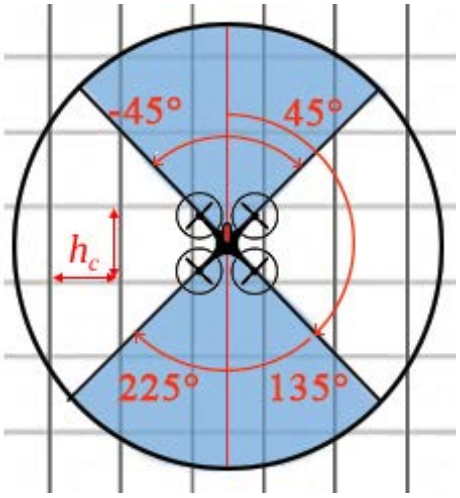

FIGURE 2. Possible UAV flight directions in the *Greedy* mode.

D. COOPERATIVE PROBABILITY MAP UPDATE

To improve efficiency, when the distance between UAVs is smaller than d_r they can communicate the real-time search and position data and share their probability map with their neighbors in order to update it using the fusion method. However, depending on the flight mode, each drone can have two different kinds of connections:

i) a temporary connection in *Greedy* mode, which will be created with other neighbors, where $\mathcal{N}_s^{(k)}(t)$ denotes the set of temporary neighbors, which must be updated whenever the UAV moves,
ii) a long connection in *LoPSO* mode, will be established with other UAVs, where $\mathcal{N}_l^{(k)}(t)$ denotes the set of long neighbors, which must not be modified even if the UAV moves.

The communication-based neighbor of U_k is typically a group of neighbors within a specified radius of d_r of such a drone's position, which may be defined as:

$$\mathcal{N}_\varphi^{(k)}(t) = \left\{ j, \parallel U_k, U_j \parallel \leqslant d_r \right\}, \tag{22}$$

where φ equals either l or s representing long neighbors and temporary neighbors, respectively, and $\parallel \cdot, \cdot \parallel$ indicates the Euclidean distance, as in Fig.1.

Also, because the probability map, which contains all probability information, is large in size and communication bandwidth is limited, the UAV only communicates the probability map with neighbors in the same sub-swarm who have established a long connection [26].

Moreover, either in *LoPSO* mode or in *Greedy* mode, while maintaining its local probability map, U_k must also maintain a local plume density map $\Psi^k(t)$ which represents the value of density per cell at time t, defined as:

$$\Psi^{(k)}(t) = \left\{ \psi_i^{(k)}(t), i = 1..M \right\}. \tag{23}$$

The process of updating the local probability map saved by U_k is divided into two steps:

- The first step is the process by which the UAV updates its local probability map individually, and the $\psi_i^{(k)}(t)$ should be updated by:

1. $\psi_i^{(k)}(t) = GetPolVal(i, t)$,

2. $$\psi_i^{(k)}(t) = \max_{n \in \mathcal{N}_\varphi^{(k)}(t)} \left\{ \psi_i^{(k)}(t), \psi_i^{(k)}(t-\tau), \psi_i^{(n)}(t-\tau) \right\}. \tag{24}$$

- The second step is to update the local probability map through information sharing and cooperation between drones. Whenever the U_k flies over the region that some of its neighbors are covering, i.e., $\{U_n\}_{n \in \mathcal{N}_\varphi^{(k)}(t)}$, U_k sends its local $P^{(k)}(t)$ and $\Psi^{(k)}(t)$ to its neighbors and receives the plume density maps and probability maps of the corresponding neighbor. Once $Z_j^{(k)}(t) = 1$, it updates its probability of detecting pollutants calculated in (21) by averaging all of its neighbors' probabilities with its own as follows:

$$P_{i,j}^{(k)}(t) = \begin{cases} P_{i,j}^{(k)}(t-\tau), & Z_j^{(k)}(t) = 0 \\ \dfrac{P_{i,j}^{(k)}(t-\tau) + \sum_{n \in \mathcal{N}_\varphi^{(k)}(t)} P_{i,j}^{(n)}(t-\tau)}{\left| \mathcal{N}_\varphi^{(k)}(t) \right| + 1}, & Z_j^{(k)}(t) = 1 \end{cases} \tag{25}$$

E. LoPSO WITH ADAPTIVE INERTIAL WEIGHT

The PSO algorithm is based on studies into the cooperative foraging behavior of birds. The basic concept is to let a swarm of particles roam a solution space in search of a solution that is close to ideal.

Each swarm particle has a location vector as well as a velocity vector. The fitness value of the current location is determined using a specified objective function while the particle velocities and positions are updated repeatedly. The particles will eventually drift towards an ideal solution after a given number of repetitions.

The particles' individual movements are based on personal information as well as the accumulation of knowledge of others. Each particle moves and cognitively interacts with the other particles in order to find the global ideal solution. These movements are precisely guided by their fitness values, which determine their optimal location as well as that of the swarm. In our suggested model, the PSO method's particles move according to comparable rules to those that control UAV movement [27].

In the *LoPSO* method we propose, all UAVs start off flying in *Greedy* mode until one U_k detects a higher pollutant level $\psi_i^{(k)}(t)$ in cell C_i. For this, we set a predefined threshold μ_p in our simulated environment: if a pollutant measurement result exceeds the threshold, it is a detection; if not, it is a non-detection. This U_k spreads the information detection about the pollutant and, depending on its $\mathcal{N}_s^{(k)}(t)$ neighbors, builds a local sub-swarm when the number of neighbors exceeds three and runs the *LoPSO* algorithm. All members of this sub-swarm alter their flight heights to the same as U_k.

Note that if different UAVs detect multiple pollution measurements that exceed the specified threshold μ_p simultaneously during the exploration stage, They will each try to build a sub-swarm to start the LoPSO stage. In this scenario, we can have multiple sub-swarms that search for the source simultaneously and independently.

If the source is correctly discovered or the time validation has elapsed, the sub-swarm shall break up and all UAVs in such sub-swarm should alter their flight heights to the origin heights and enter search *Greedy* mode.

The position and velocity of U_k are updated based on *LoPSO*, respectively, in each iteration cycle, as follows:

$$\mathcal{V}_{\nu}^{(k)}(t) = \lambda^{(k)}(t-\tau)\mathcal{V}_{\nu}^{(k)}(t-\tau) + r_1 c_1 \mathcal{V}_{l,\nu}^{(k)}(t-\tau) + r_2 c_2 \mathcal{V}_{g,\nu}^{(k)}(t-\tau), \tag{26}$$

where

$$\mathcal{V}_{l,\nu}^{(k)}(t-\tau) = \frac{\delta_{\nu}^{(k)}(t-\tau) - \chi_{\nu}^{(k)}(t-\tau)}{\tau}, \tag{27}$$

$$\mathcal{V}_{g,\nu}^{(k)}(t-\tau) = \frac{\Delta_{\nu}^{(k)}(t-\tau) - \chi_{\nu}^{(k)}(t-\tau)}{\tau}, \tag{28}$$

and

$$\chi_{\nu}^{(k)}(t) = \chi_{\nu}^{(k)}(t-\tau)_{+}\mathcal{V}_{\nu}^{(k)}(t-\tau)\tau, \tag{29}$$

with ν equals either x or y, c_1 and c_2 are the individual and social cognitive coefficients of the UAV, r_1 and r_2 are random values referring to the acceleration coefficients uniformly distributed between 0 and 1 [27]. Further, $\delta^{(k)}(t)$ denotes the best position of U_k, and $\Delta^{(k)}(t)$ as the best position of the sub-swarm.

Let's defined the following fitness function

$$g\left(\chi^{(k)}(t)\right) = \frac{(\psi_i^{(k)}(t))^{P_{i,j}^{(k)}(t)} - 1}{(\psi_i^{(k)}(t))^{P_{i,j}^{(k)}(t)} + 1} + 1. \tag{30}$$

In this paper, the objective is to achieve the global optimal by maximizing the fitness function.

It is obvious that the higher the source existence probability with a very high level of plume density, the better the fitness. As a result, the best position of U_k, i.e., $\delta^(k)(t)$, is defined as the position that corresponds to the fitness function's highest value up until time t.

$$\delta^{(k)}(t) = \arg\max_{t} \left\{ g\left(\chi^{(k)}(t)\right), g\left(\delta^{(k)}(t-\tau)\right) \right\}. \tag{31}$$

Furthermore, once the sub-swarm is constructed, $\delta^{(k)}(t)$ is initialized at the position of U_k. Whereas the global best position of the sub-swarm $\mathcal{N}_l^{(k)}(t)$ at a given time t, i.e., $\Delta^{(k)}(t)$ assumed to be the probable position of the source, can be found by depending on the local best positions $\delta^{(k)}(t)$ as

$$\Delta^{(k)}(t) = \arg \max_{q \in \left\{k, \mathcal{N}_l^{(k)}(t)\right\}} \left\{ g(\delta^{(q)}(t)) \right\}. \tag{32}$$

To achieve a balance between global and local exploration by the sub-swarm, the value of the inertia weight $\lambda^{(k)}(t)$ which assures convergence will be determined by the speed of evolution and the degree of population aggregation.

Also, when the function fitness is less than a specific threshold μ_o, $\lambda^{(k)}(t)$ stays large and a global search is performed, and when the fitness is greater than the threshold, $\lambda^{(k)}(t)$ performs an adaptive nonlinear decline to do a fine search by continually approaching the likely source point.

Taking advantage of the sigmoid function which has a strong capacity for nonlinear approximation and its extreme value is between 0 and 1, we introduce it to adaptively update the inertia weight value, then the specific expression is:

$$\lambda^{(k)}(t) = \begin{cases} \frac{2}{1+\exp\left(\frac{-1}{g\left(\Delta^{(k)}(t)\right)}\right)} - 1.4, & g\left(\Delta^{(k)}(t)\right) > \mu_o \\ 0.9, & g\left(\Delta^{(k)}(t)\right) \leqslant \mu_o. \end{cases} \tag{33}$$

III. SEARCH STRATEGY

The suggested approach uses a search strategy based on UAV cooperation and a probabilistic method that attempts to solve the problem of determining the pollution source position. While the UAVs use two search modes *Greedy* and *LoPSO* to track the growing concentration of the pollutant, they are further assisted by a heuristic source position in order to adjust the computed directions to focus the search on areas with higher pollution concentrations.

For the purpose of designing the algorithm, the complexity of finding the polluting source has been reduced to two dimensions. The development potential is unaffected by this lowering. In reality, because the vertical propagation impact of air pollution is substantially smaller than the horizontal propagation effect [33], the equations may be simplified and the search area effectively reduced.

During the search, the measurements of the pollution level of air given by the sensors are analyzed to decide whether there is a detection of pollutants, which will be used by the probabilistic component. At each UAV, the search task is separated into three parts:

- *Collecting* measurements with the UAV's sensors;
- *Exchanging* information and fusing detection finding;
- *Coordinating* movement depending on the flight mode.

In our approach, in order to pilot the drones cooperatively and achieve the intended goal, we divided the algorithm into two parts: identification and tracking of the plume, followed by the localization of the polluting source.

A. PLUME TRACING ALGORITHM

The detection and tracing of the plume are based on the notion of pollutant detection heuristics, namely the gradient algorithm, which needs at least two spatially separated measurements in order to look for the highest pollution concentration levels in a region.

In this context, to increase the efficiency of the localization of the pollutant plume, we deploy a *Greedy* strategy to accelerate the process of convergence of the probability map of each drone during the search task.

At this point, the algorithm is divided into two phases: (i) the searching phase, wherein the drones look for higher

pollution measurement values, and (ii) the exploration phase, wherein the drones fly around the area until one of the following conditions is met: the drones have covered the entire area and there are no sources of pollutants on the sides and the flight duration has expired, or a new higher value is found.

As described in Algorithm 2, each drone U_k collects the pollution measurement value at its current position C_k when it makes its initial movement, and delivers it to the Algorithm 4, if the value $\psi_i^{(k)}(t)$ is greater than a threshold μ_p, otherwise, it compares to the previous measurement $\psi_i^{(k)}(t-\tau)$. If the measurement variation $\Delta\psi_i^{(k)}(t)$ is positive and greater than a threshold of μ_s (increasing), determined in accordance with the sensor measurement, the direction of the UAV's motion is in *Greedy* mode with $Z_i^{(k)}(t) = 1$ and toward an area where the amount of air pollution is higher. As a result, the exploration phase begins, which is mainly based on two techniques: the drone follows a direction ahead towards the cell with a high pollution concentration level between $[-45°, 45°]$ degrees. Otherwise, the UAV flies in the wrong direction if the measurement variance decreases. However, the drone calculates the direction in reverse orientation to move back toward the cell with a high pollution concentration level between [135°, 225°] degrees and continues the search Fig. 2.

Once a pollution hotspot has been identified by one UAV, this later will create a sub-swarm alliance through neighborhood communication and interaction, then the *LoPSO* search algorithm will be deployed to coordinate the search for the phase source tracking Fig. 3.

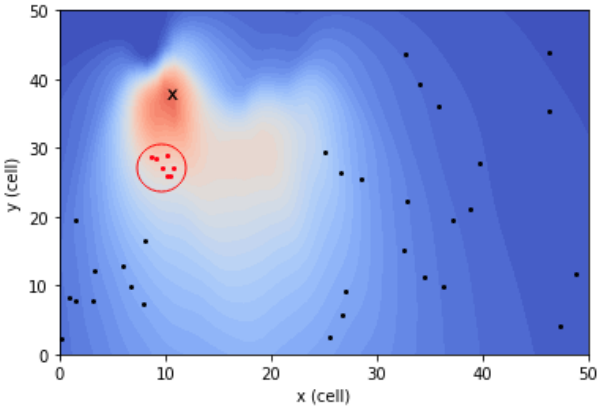

FIGURE 3. Plume tracing: The blacks are Greedy UAVs, while the reds are a sub-swarm in LoPSO mode that formed after detecting the plume.

B. POLLUTANT SOURCE LOCALIZATION

At this point, the Algorithm 4 is used at this step to focus the search on the regions where the source is most likely to be found, depending on the plume's higher concentration level.

The *LoPSO*, on the other hand, has two phases: the exploration phase and the exploitation phase. During the exploration phase, the formed sub-swarm seeks a higher overall pollution value; this phase ends when a high local level of pollution is identified. When the pollutant concentration level reaches its maximum value, the drones from the sub-swarm begin to exploit the region containing the identified plume in order to confirm or deny the location C_s of the specific source which contains the higher concentration level using the probability map. The C_s cell, which is the most likely location on the probabilistic map, is the focus of this phase, during which each drone flies in cooperation with others in the same sub-swarm around this cell. The whole process of our approach can be summed up in algorithms Algorithm 1-5.

The main algorithm used by the base station (BS) to start the cooperative search is described in Algorithm 1 function. After providing the input settings, all UAVs launch and start searching till the search period is over. Until the polluting source is found, the search is still ongoing. All drones will be warned to halt the search in this scenario. To gather data about the source discovered, the BS continuously listens to all drones.

In each UAV, the Algorithm 2 describes plume searching, coordination updating, and information merging. The basic process is that all drones must first begin searching for the pollutant plume in *Greedy* mode by running the Algorithm 5, provided that the base station has not yet indicated that the search is over and there is sufficient energy to do so.

To validate a pollutant's existence and update their maps of probability and concentration of pollutants in the area, the UAVs collect detection information from their sensors using the *readpollution* function on the hovered cell. They then merge and share information with other nearby drones provided by the Algorithm 3, which builds the long neighbors. When a sub-swarm integration request is submitted when the flight mode is *Greedy*, the UAV will accept the request, switch to *PSO*, send an acceptance message, integrate the sub-swarm, and then run the Algorithm 4.

If the UAV is in the *Greedy* flight mode and a plume is found in the hovering cell, it switches to the *PSO* flight mode and launches the Algorithm 4 by forming a sub-swarm based on the Algorithm 3. When the source is confirmed, the base station is notified of the source position, releases the sub-swarm, and enters the greedy flying mode. If the flying mode is *PSO*, it is essential to update the movement information in accordance with (26) and (29) as long as there is no release request.

IV. EXPERIMENTS AND RESULTS

In this section, simulation results are provided in order to demonstrate the effectiveness and reliability of our proposed approach and to assess its functioning as realistically as possible. Several simulation experiments were performed to confirm the behavior of the proposed approach while fixing the values of their parameters to those listed in Table 3. Some statistical results were obtained by simulating the drones and a source of pollution on a platform with a different number of drones deployed at each test. In our simulation, the statistical findings from every experiment are the mean values based on

Algorithm 1 Base Station Monitoring Program

Function `Monitoring` $(\Omega, N_u, D_{\max}, \tau)$ **:**
- `/* lunched on the base station */`
- **init**
  - $search \leftarrow True$
  - $B_{0,0}(True)$ `/* alert all UAVs that the search in ongoing */`
  - $C_s \leftarrow random()$
  - $t \leftarrow 0$
  - $n \leftarrow 0$
- **for** $i \leftarrow 1$ **to** N_u **do**
  - `/* order all UAVs to start the search */`
  - $Searching(N_u, i, N_x, N_y, C_s, t, \tau, D_{\max}, \mu_p)$
- **while** $search$=$True$ *and* $t <= D_{\max}$ **do**
  - **for** $i \leftarrow 1$ **to** N_u **do**
    - `/* sensing a exist transmission data from each UAV */`
    - $R_i(n, I)$
    - **if** n=4 **then**
      - `/* The data obtained is the location of the source */`
      - $search \leftarrow$ **False**
      - `/* all UAVs must be alerted to halt the search */`
      - $B_0(0, False)$
      - `/* return the source position */`
      - $C_s \leftarrow I$
  - $t \leftarrow t + \tau$
- **return** C_s

Algorithm 2 UAV Pollutant Source Search

Function `Searching` $(N_u, k, N_x, N_y, C_s, t, \tau, D_{\max}, \mu_p)$ **:**
- `/* Achieved in each UAV */`
- **init**
  - $\mathcal{V}^{(k)}(t)$, $\chi^{(k)}(t)$, E_t, E_k
  - $P^k(t) \leftarrow \left\{P^{(k)}_{i,j}(t) = \frac{1}{N_x \times N_y}, i = 1..N_x, j = 1..N_y\right\}$
  - $\Psi^k(t) \leftarrow \left\{\psi^{(k)}_{i,j}(t) = 0, i = 1..N_x, j = 1..N_y\right\}$
  - $SSwHead \leftarrow$ **False**
  - $F_k \leftarrow$ "$Greedy$"
  - $Search \leftarrow$ **True**
  - $n \leftarrow 0$
  - $t \leftarrow 0$
  - $\mathcal{N}^{(k)}_l(t) \leftarrow \{\phi\}$
  - $\mathcal{N}^{(k)}_s(t) \leftarrow \{\phi\}$
- `/* collect the pollutant concentration in the current location */`
- $\theta_1 \leftarrow GetPolVal(i, t)$
- **while** $t < D_{\max}$ and $E_k > E_t$ and $Search \neq$ False **do**
  - $\theta_2 \leftarrow GetPolVal(i, t)$
  - **if** $F_k =$ "$Greedy$" **then**
    - `/* flight mode is greedy: exploration stage */`
    - **if** $\theta_2 > \mu_p$ **then**
      - `/* detection of a pollutant */`
      - $Z^{(k)}_i(t) \leftarrow$ **1**
      - **update** $P^{(k)}(t)$, $\Psi^{(k)}(t)$ `/* according to (21), (24) and (23) */`
      - $B_k(1, P^{(k)}(t))$
      - `/* share the probability map with neighbors */`
      - $B_k(2, \Psi^{(k)}(t))$
      - `/* share the plume density map with neighbors */`
      - $SSwHead \leftarrow$ **True**
      - `/* take the heading role */`
      - $C_s \leftarrow LoPSO(N_u, \Omega, C_s, k, t, \tau, SSwHead)$
      - `/* run the LoPSO mode */`
      - **if** $C_s \neq$ False **then**
        - `/* case where one source found */`
        - $B_k(5, C_s)$
        - `/* share the source location */`
        - $SSwHead \leftarrow$ **False** `/* quit the sub-swarm header role */`
    - **else**
      - `/* no plume detected */`
      - $\Delta\theta \leftarrow \theta_2 - \theta_1$
      - $\mathcal{N}^{(k)}_l(t) \leftarrow Greedy(N_u, C_s, k, t, \Delta\theta, \mu_s, i)$
        - `/* retrieve the list of long neighbors */`
  - **else**
    - `/* in case the flight mode is LoPSO */`
    - $C_s \leftarrow LoPSO(N, \Omega, C_s, k, t, \tau, SSwHead)$
  - $t \leftarrow t + \tau$
  - $E_k \leftarrow E_k - 1$
  - $R_0(n, I)$
  - `/* sensing exist transmission */`
  - **if** $n = 0$ **then**
    - `/* exist transmission from BS */`
    - $Search \leftarrow I$
  - $\theta_1 \leftarrow \theta_2$
- **return** $True$

$D_{\max} = 1000$ independent tests. The suggested approach also presupposes the following circumstances:

- The intended environment is confined to a horizontal rectangle;
- The pollutant is only released from one source;
- The location of the source in the environment is randomly selected in the algorithm's initial phase;
- Without knowledge of the polluted plume, the drones begin the mission at random locations in the intended environment;
- Each drone has a sensor for measuring pollutants;
- It is possible to estimate the wind vector in the horizontal plane;
- Variations in winds do not significantly affect the positioning (navigation) control of drones;
- Vehicle flight time, in-vehicle memory, and processing capacity are limited.

Fig.4 depicts how the number of UAVs deployed affects how long it takes to find the source of the pollution for each of the four search methods. We can see that the validation time of the source in LoPSO quickly decreases as the number of UAVs increases, compared to other methods. This is due to the fact that once the plume is detected by one UAV, the local swarm built allows it to be quickly validated by the increasing

Algorithm 3 Neighbors in Close Proximity

Function `NearbyNeighbors` (k,t,N_u) **:**
  /* build list of neighbors */
  if $F_k =$ "*Greedy*" **then**
    /* flight mode is greedy */
    $\mathcal{N}_s^{(k)}(t) \leftarrow \{\phi\}$
    $B_k(0,$ "*Who*"$)$
    /* broadcast a discovery message */
    for $j \leftarrow 1$ **to** N_u **do**
      $R_j(n,I)$
      /* sensing exist potential transmission */
      if $n = 0$ and $I =$ True **then**
        /* a UAV neighbor was found */
        $\mathcal{N}_s^{(k)}(t) \leftarrow \mathcal{N}_s^{(k)}(t) \cup \{j\}$
        /* build list of short neighbors */
    return $\mathcal{N}_s^{(k)}(t)$

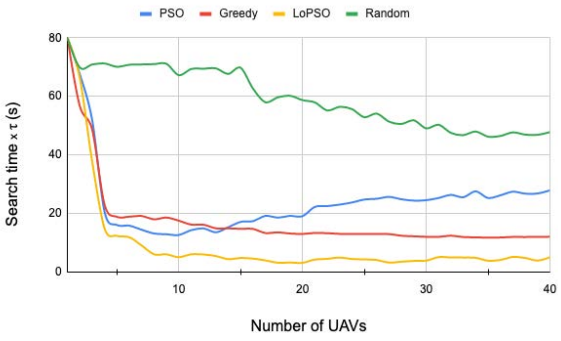

FIGURE 4. Time taken to confirm a source's location.

number of detections carried out by the drones in this local swarm. The statistical results permit this observation to be verified, as shown in Table 4. In fact, as the size of the drone swarm system increases, the total movement of all drones increases rapidly, allowing rapid detection of the plume due to the wide dispersion of the drones. This results in the formation of sub-swarms which will quickly and efficiently exploit the reduced area where the feather is detected and therefore a shorter search time.

Fig. 5 illustrates the average energy usage as a function of the number of UAVs. We consider the amount of energy used for both communication and movement. These results show that the number of UAVs used increases power consumption over time and that the proposed *LoPSO* strategy uses, on average, 50% less power than previous methods. In fact, as the drones travel to the next cell at the same speed, the energy needed for mobility will continuously rise.

However, with *LoPSO*, the swarm that forms determines the pace of the drones, so the UAVs' speed is not constant.

Algorithm 4 Local PSO Mode

Function `LoPSO` $(N_u,\Omega, C_s, k, t, \tau, SSwHead, \mu_o, t_{PSO})$**:**
  init
    $t_k \leftarrow t$
    $F_k \leftarrow$ "*LoPSO*"
    Search ←**True**
    $(\delta^{(k)}(t), \Delta^{(k)}(t), \Delta_{\max}) \leftarrow (\chi^{(k)}(t), \chi^{(k)}(t), \chi^{(k)}(t))$
  foreach $j \in \mathcal{N}_l^{(k)}(t)$ **do**
    $R_j(n,I)$ /* Sensing exist transmission */
    if $I =$"Free" and $n = 0$ **then**
      /* leave the sub-swarm */
      $F_k \leftarrow$ "*Greedy*" /* switch to greedy mode */
  if *SSwHead* = True **then**
    /* The current UAV is the sub-swarm header */
    $\mathcal{N}_l^{(k)}(t) \leftarrow \{\phi\}$
    $\mathcal{N}_s^{(k)}(t) \leftarrow NearbyNeighbors(k,t,N)$
    /* collect the list of short neighbors */
    foreach $j \in \mathcal{N}_s^{(k)}(t)$ **do**
      $B_k(0,$ "*Join*"$)$
      /* invite close neighbors to join the sub-swarm */
      $R_j(n,I)$ /* collect responses from nearby neighbors */
      if $I =$True and $n = 0$ **then**
        $\mathcal{N}_l^{(k)}(t) \leftarrow \mathcal{N}_l^{(k)}(t) \cup \{j\}$ /* build the sub-swarm */
    if $\left|\mathcal{N}_l^{(k)}(t)\right| \geqslant 3$ **then**
      foreach $j \in \mathcal{N}_l^{(k)}(t)$ **do**
        $B_k(3, \mathcal{N}_l^{(k)}(t))$ /* share the sub-swarm list */
      while *Search* ← True and $t - t_k \leqslant t_{PSO}$ **do**
        $\theta \leftarrow$ **GetPolVal(i,t)**
        $Z^{(k)}(t)_i \leftarrow \mathbf{1}$
        compute $\delta^{(k)}(t)$, $\Delta^{(k)}(t)$,$\mathcal{V}^{(k)}(t)$, $\chi^{(k)}(t)$
          /* according (31),(32),(26), and (29) */
        $B_k(5, \Delta^{(k)}(t))$/* share the global best position of the sub-swarm */
        update $P^k(t)$, $\Psi^k(t)$ /* according (21), (24) and (23) */
        Move to $\chi^{(k)}(t)$
        if $g(\Delta^{(k)}(t)) \geq g(\Delta_{\max})$ **then**
          $\Delta_{\max} \leftarrow \Delta^{(k)}(t)$
        if $\left\|\Delta^{(k)}(t), \Delta_{\max}\right\| > d_r$ **then**
          $F_k \leftarrow$ "*Greedy*";
          foreach $j \in \mathcal{N}_l^{(k)}(t)$ **do**
            $B_k(0,$ "*Free*"$)$/* alert all to quit the sub-swarm */
          $\mathcal{N}_l^{(k)}(t) \leftarrow \{\phi\}$
          return $\Delta_{\max}$ /* best location found */
        else
          $t \leftarrow t + \tau$
      if $t - t_k > t_{PSO}$ **then**
        $F_k \leftarrow$ "*Greedy*" /* change to greedy */
        foreach $j \in \mathcal{N}_l^{(k)}(t)$ **do**
          $B_k(0,$ "*Free*"$)$
        $\mathcal{N}_l^{(k)}(t) \leftarrow \{\phi\}$
        $\mathcal{N}_s^{(k)}(t) \leftarrow \{\phi\}$
        /* empty all lists of neighbors */
        return $(-1,-1)$/* no source found */
    else
      /* UAV fails to build the sub-swarm */
      foreach $j \in \mathcal{N}_l^{(k)}(t)$ **do**
        $B_k(0,$ "*Free*"$)$/* notify all neighbors to breaking free from the sub-swarm */
      $F_k \leftarrow$ "*Greedy*"/* back to greedy mode */
      $\mathcal{N}_l^{(k)}(t) \leftarrow \{\phi\}$
      $\mathcal{N}_s^{(k)}(t) \leftarrow \{\phi\}$
      /* empty all lists of neighbors */
      $H_k \leftarrow k$/* back to original flight height */
  else
    /* the UAV is a simple sub-swarm member */
    while *Search*= True and $t - t_k \leqslant t_{PSO}$ **do**
      $\theta \leftarrow$ **GetPolVal(i,t)**
      $Z_i^{(k)}(t) \leftarrow \mathbf{1}$
      compute $\delta^{(k)}(t)$,$\Delta^{(k)}(t)$, $\chi^{(k)}(t)$,$\mathcal{V}^{(k)}(t)$
      update $P^{(k)}(t)$, $\Psi^{(k)}(t)$/* using (21), (24) and (23) */
      Move to $\chi^{(k)}(t)$
      $B_k(5, \delta^{(k)}(t))$/* share the individual best position to the sub-swarm */

Algorithm 5 Greedy Mode

Function Greedy (N_u,C_s,k,t,$\Delta\theta$, μ_s,i) :
  if $\Delta\theta > \mu_s$ **then**
    /* plume detected */
    $Z^{(k)}(t)_i \leftarrow$**1**
  Update $P^{(k)}(t)$/* according (21) and (24) */
  Update $\Psi^{(k)}(t)$/* according (23) */
  $B_k(1, P^{(k)}(t))$/* share probability map with neighbors */
  $B_k(2, \Psi^{(k)}(t))$/* share plume density map with neighbors */
  $\mathcal{N}_s^{(k)}(t) \leftarrow NearbyNeighbors(k, t, N_u)$/* get short neighbors list */
  Move to next cell
  foreach $j \in \mathcal{N}_s^{(k)}(t)$ **do**
    $R_j(n, I)$
    if I="Join" and $n = 0$ **then**
      /* sensing if there is a request to join a sub-swarm */
      $B_k(0, True)$ /* accept to join the invitation to the sub-swarm */
      $F_k \leftarrow$ "$LoPSO$" /* change flight mode to LoPSO */
      $H_k \leftarrow j$/* take U_j's flight height */
      break
    if $n = 3$ **then**
      $\mathcal{N}_l^{(k)}(t) \leftarrow I$
      return $\mathcal{N}_l^{(k)}(t)$ /* collect list of long neighbors */
    if $n = 0$ and I="Who" **then**
      /* sensing a message of exploration */
      $B_k(0, True)$ /* declare the existence of the UAV */
  return $\{\phi\}$

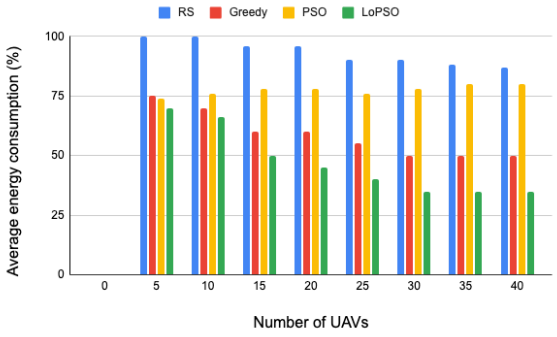

FIGURE 5. Energy consumption vs number of UAVs.

Additionally, on the one hand, the drones only transmit information when it is absolutely essential thanks to the messaging technique described, which implies reducing the volume of messages exchanged between drones. On the other hand, as shown in Table 4, the size of the sub-swarms built

TABLE 3. Parameters settings.

Parameter	*Value*
N_u	40
$v_{\max}^{(k)}(t)$	15 m/s
$v_{\min}^{(k)}(t)$	0.5 m/s
$v_{\text{avg}}^{(k)}(t)$	10 m/s
μ_s	0.50
μ_o	0.99
μ_p	0.80
N_x	50
N_y	50
h_c	100 m
H	40 m
r_1	$]0, 1[$
d_r	4
r_2	$]0, 1[$
c_1	$[1.1, 1.4]$
τ	10 s
c_2	$[1.1, 1.4]$
$\lambda^{(0)}$	1.2
$D_{\max}$	1000 s
t_{PSO}	200 s
σ_x	$1 \times 10{-}3$ m/s
σ_y	$1 \times 10{-}3$ m/s
σ_z	$1 \times 10{-}3$ m/s
ω	5 m/s
ω_{set}	2.7×10^{-3} m/s
W_{dep}	5×10^{-3} m/s
Q	$[1.1, 2.5, 1.6, 1.6] \times 10^3$ kg/s

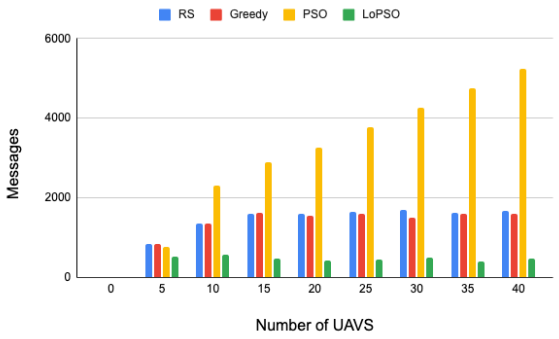

FIGURE 6. Messages exchanged vs the number of UAVs.

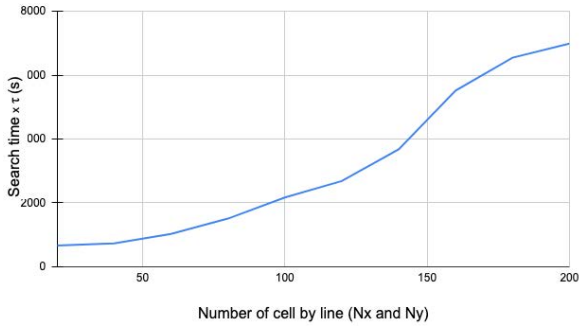

FIGURE 7. Required time to locate the source position for different search area size.

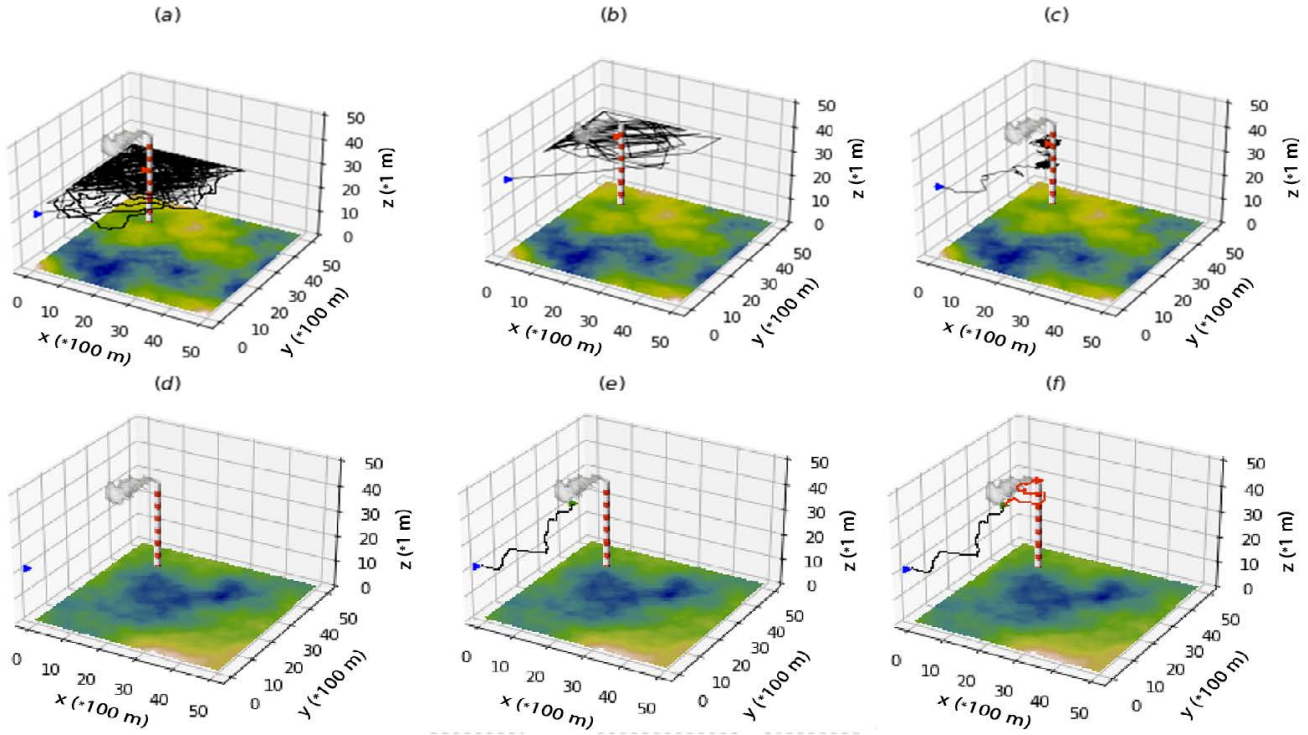

FIGURE 8. Search path of one UAV in different modes: (a) Random search, (b) PSO mode, (c) Greedy mode,(d) search start in LoPSO mode (e) plume detection, and (f) source localization.

TABLE 4. Average confirmation time, fitness function max, validation probability max, average search time, and number of neighbors under different UAVs number.

N_u	$Conf$	$g_{\max}$	$P_{\max}$	ST	$Neig$
5	7.352	1.152	0.997	4016.95	3
10	5.055	1.123	0.997	1449.35	4
15	4.802	1.137	0.998	1077.95	4
20	4.142	1.127	0.998	1009.25	5
25	3.916	1.136	0.999	923.50	5
30	3.889	1.165	0.999	815.85	5
35	3.867	1.207	0.999	805.80	6
40	3.121	1.232	0.999	735.50	6

does not have large sizes (between 3 and 6), which reduces the quantity of messages exchanged locally. Consequently, since communication may only occur inside the same swarm, this decreases the overall number of messages exchanged between the different UAVs, as shown in Fig. 6.

In the suggested method, the environment is discretized as a grid, and the search duration varies with grid size (number of cells). Fig. 7 shows that the computation time increases as the search area size grows, implying an increase in computational complexity. Indeed, there are two periods of time during research: exploration and exploitation. Since the UAVs will be dispersed throughout a sizable area during the first phase, the exploration time will vary depending on the area size, while the exploitation time will stay about the same.

The drones cooperated on two levels: individually with drones nearby and collectively with drones belonging to the same sub-swarm, ensuring quick and efficient convergence of the algorithm in terms of time and energy. Indeed, the algorithm was built using both exploration and exploitation strategies. The use of gluttony kept exploration in the most promising locations, while the LoPSO metaheuristics allowed fewer drones to examine smaller areas by following the increasing gradient of pollutants and therefore decreased paths.

As shown in Fig. 8, the source of the pollutant has been found by the different algorithms in the majority of cases, but with varying trajectory lengths: (a) *RS*, (b) *PSO*, (c) *Greedy*, and (d, e, and f) *LoPSO*. As depicted, the trajectories are lengthy, implying a considerable amount of search time. Nonetheless, as compared to other methods, the total trajectory in *LoPSO* is shorter. Indeed, the black trajectory shown in (e) is the first part of the search using the *Greedy* method and prioritizing tracking of high levels of pollutant concentration. The launch of *LoPSO* (f) to locate the source using a small number of drones, which speeds up the detection by tracking the red trajectory, occurs after discovering the plume at the location indicated by the green triangle.

Fig. 9 depicts a comparison of the plume density map versus the search time duration for one of the swarm's UAVs at various intervals. it can be seen that the probability of the source is quite high in a very small place, but it is almost zero outside of this small area. The pollutant source was found in cell C_{1911} (i.e., column = 11, row = 38) at time $t = 735$ s, which presented the concentration and probability of highest pollutants ($\psi = 1542\ \mu g/m^2$, $P = 0.9998$).

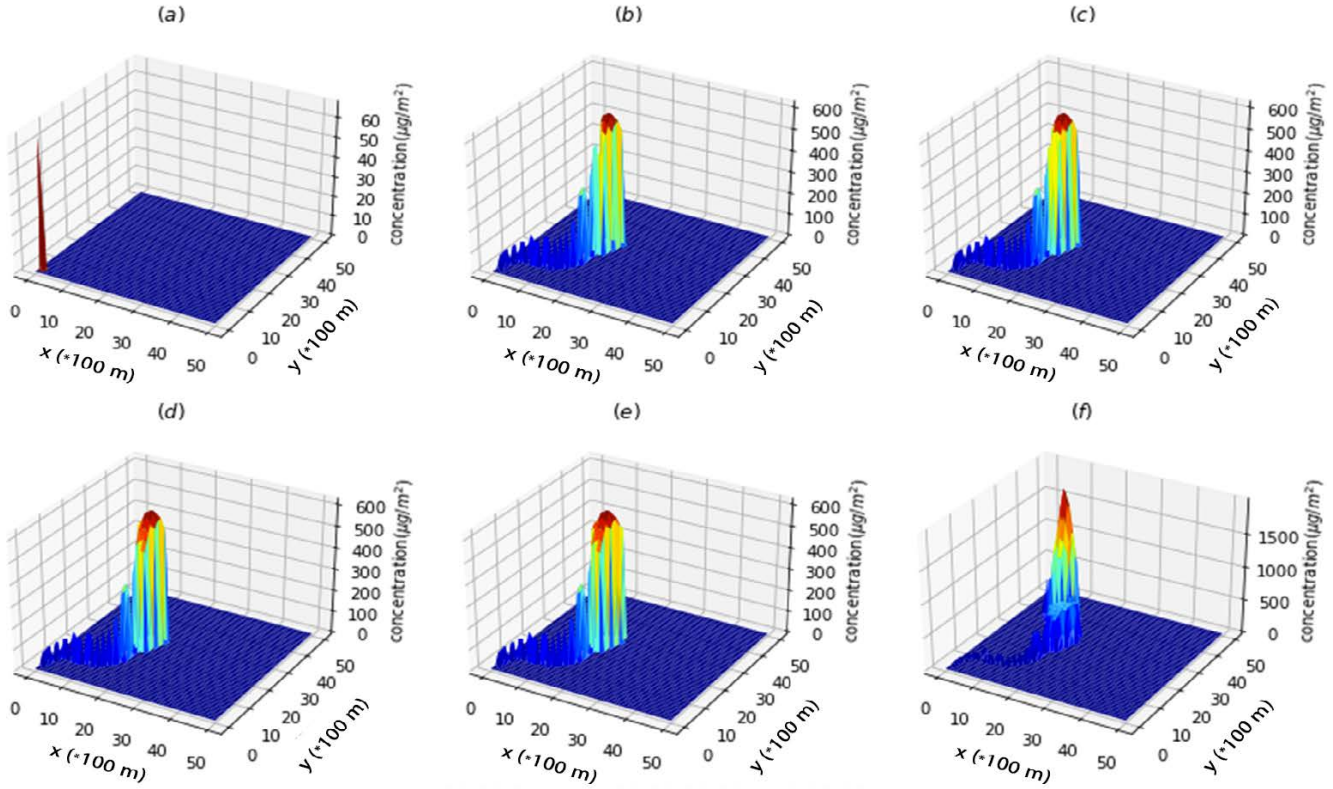

FIGURE 9. Plume map tracing for a fixed UAV for $d_r = 8$ cells. (a) $t = 0s$. (b) $t = 220s$. (c) $t = 441s$. (d) $t = 537s$. (e) $t = 672s$. (f) $t = 736s$.

V. CONCLUSION

This paper provided a method for locating the source of air pollution across a vast region by prioritizing the most contaminated locations utilizing autonomously cooperative drones. The suggested approach employs a two-phased equidistributed search based on the Bayesian process model. In order to rapidly detect the polluting plume during the exploration phase, it employs a search based on the *Greedy* algorithm. In this step, the drones cover different cells of the search area by moving toward those containing a high concentration level of the pollutant. The *LoPSO* meta-heuristic algorithm and the Bayesian approach are coupled during the exploitation phase to produce a thorough and in-depth probabilistic map of the area, enabling the localization of the source. In simulated tests, the proposed approach was compared to three others in real-time using a dispersion-advection plume model that simulated the real world and a modeled pollution source with significant turbulence. The findings demonstrated the algorithm's excellent performance and robustness, as the drones were able to locate the pollution source in a variety of environmental situations. Future research directions will be focused on examining the benchmarking capabilities of LoPSO and its capacity to resolve further challenging optimization problems, such as adding a wind model from collected wind data to strengthen the proposed model while trying to conduct experimental experiments using real quadcopters in a real polluted outdoor environment. Also, using LoPSO jointly with a reinforcement learning strategy is a promising idea to be investigated.

REFERENCES

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

[2] L. Hashemi-Beni and A. A. Gebrehiwot, "Flood extent mapping: An integrated method using deep learning and region growing using UAV optical data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 2127–2135, 2021.

[3] *Ambient Air Pollution: A Global Assessment of Exposure and Burden of Disease*, World Health Org., Geneva, Switzerland, 2016.

[4] T. Benmarhnia, "Linkages between air pollution and the health burden from COVID-19: Methodological challenges and opportunities," *Amer. J. Epidemiol.*, vol. 189, no. 11, pp. 1238–1243, Nov. 2020.

[5] S. C. van der Zee, P. H. Fischer, and G. Hoek, "Air pollution in perspective: Health risks of air pollution expressed in equivalent numbers of passively smoked cigarettes," *Environ. Res.*, vol. 148, pp. 475–483, Jul. 2016.

[6] T. F. Villa, F. Salimi, K. Morton, L. Morawska, and F. Gonzalez, "Development and validation of a UAV based system for air pollution measurements," *Sensors*, vol. 16, no. 12, p. 2202, 2016.

[7] V. Lambey and A. D. Prasad, "A review on air quality measurement using an unmanned aerial vehicle," *Water, Air, Soil Pollut.*, vol. 232, no. 3, pp. 1–32, Mar. 2021.

[8] T. F. Villa, F. Gonzalez, B. Miljievic, Z. D. Ristovski, and L. Morawska, "An overview of small unmanned aerial vehicles for air quality measurements: Present applications and future prospectives," *Sensors*, vol. 16, no. 7, p. 1072, 2016.

[9] N. M. Yungaicela-Naula, Y. Zhang, L. E. Garza-Castañon, and L. I. Minchala, "UAV-based air pollutant source localization using gradient and probabilistic methods," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2018, pp. 702–707.

[10] R.-G. Li and H.-N. Wu, "Multi-robot source location of scalar fields by a novel swarm search mechanism with collision/obstacle avoidance," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 249–264, Jan. 2022.
[11] Q. Feng, C. Zhang, J. Lu, H. Cai, Z. Chen, Y. Yang, F. Li, and X. Li, "Source localization in dynamic indoor environments with natural ventilation: An experimental study of a particle swarm optimization-based multi-robot olfaction method," *Building Environ.*, vol. 161, Aug. 2019, Art. no. 106228.
[12] D. Facinelli, M. Larcher, D. Brunelli, and D. Fontanelli, "Cooperative UAVs gas monitoring using distributed consensus," in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2019, pp. 463–468.
[13] P. Li and H. Duan, "A potential game approach to multiple UAV cooperative search and surveillance," *Aerosp. Sci. Technol.*, vol. 68, pp. 403–415, Sep. 2017.
[14] Y. Yang, B. Zhang, Q. Feng, H. Cai, M. Jiang, K. Zhou, F. Li, S. Liu, and X. Li, "Towards locating time-varying indoor particle sources: Development of two multi-robot olfaction methods based on whale optimization algorithm," *Building Environ.*, vol. 166, Dec. 2019, Art. no. 106413.
[15] O. Alvear, N. R. Zema, E. Natalizio, and C. T. Calafate, "Using UAV-based systems to monitor air pollution in areas with poor accessibility," *J. Adv. Transp.*, vol. 2017, pp. 1–14, Aug. 2017.
[16] R. Kristiansen, E. Oland, and D. Narayanachar, "Operational concepts in UAV formation monitoring of industrial emissions," in *Proc. IEEE 3rd Int. Conf. Cogn. Infocommun. (CogInfoCom)*, Dec. 2012, pp. 339–344.
[17] J. Han, Y. Xu, L. Di, and Y. Chen, "Low-cost multi-UAV technologies for contour mapping of nuclear radiation field," *J. Intell. Robot. Syst.*, vol. 70, no. 1, pp. 401–410, Apr. 2013.
[18] Z. Shen, Z. He, S. Li, Q. Wang, and Z. Shao, "A multi-quadcopter cooperative cyber-physical system for timely air pollution localization," *ACM Trans. Embedded Comput. Syst.*, vol. 16, no. 3, pp. 1–23, Aug. 2017.
[19] F. Jiang, L. Dong, and Q. Dai, "Designing a mixed multilayer wavelet neural network for solving ERI inversion problem with massive amounts of data: A hybrid STGWO-GD learning approach," *IEEE Trans. Cybern.*, vol. 52, no. 2, pp. 925–936, Feb. 2022.
[20] S. G. Nurzaman, Y. Matsumoto, Y. Nakamura, S. Koizumi, and H. Ishiguro, "Biologically inspired adaptive mobile robot search with and without gradient sensing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 142–147.
[21] T. Lochmatter and A. Martinoli, "Tracking odor plumes in a laminar wind field with bio-inspired algorithms," in *Experimental Robotics* (Springer Tracts in Advanced Robotics), vol. 54. Berlin, Germany: Springer, 2009, pp. 473–482, doi: 10.1007/978-3-642-00196-3_54.
[22] P. P. Neumann, V. H. Bennetts, A. J. Lilienthal, M. Bartholmai, and J. H. Schiller, "Gas source localization with a micro-drone using bio-inspired and particle filter-based algorithms," *Adv. Robot.*, vol. 27, no. 9, pp. 725–738, 2013.
[23] Q.-V. Pham, T. Huynh-The, M. Alazab, J. Zhao, and W.-J. Hwang, "Sum-rate maximization for UAV-assisted visible light communications using NOMA: Swarm intelligence meets machine learning," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10375–10387, Oct. 2020.
[24] F. Jiang, L. Dong, K. Wang, K. Yang, and C. Pan, "Distributed resource scheduling for large-scale MEC systems: A multiagent ensemble deep reinforcement learning with imitation acceleration," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6597–6610, May 2022.
[25] F. Jiang, K. Wang, L. Dong, C. Pan, W. Xu, and K. Yang, "Deep-learning-based joint resource scheduling algorithms for hybrid MEC networks," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6252–6265, Jul. 2020.
[26] H. Saadaoui and F. E. Bouanani, "Communication and energy optimization of local PSO-assisted multi-UAVs for moving targets exploration," in *Proc. 4th Int. Conf. Adv. Commun. Technol. Netw. (CommNet)*, Dec. 2021, pp. 1–7.
[27] H. Saadaoui, F. E. Bouanani, and E. Illi, "Information sharing based on local PSO for UAVs cooperative search of moved targets," *IEEE Access*, vol. 9, pp. 134998–135011, 2021.
[28] S. Pang and J. A. Farrell, "Chemical plume source localization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 5, pp. 1068–1080, Oct. 2006.
[29] B. P. Van Milligen, P. Bons, B. A. Carreras, and R. Sanchez, "On the applicability of Fick's law to diffusion in inhomogeneous systems," *Eur. J. Phys.*, vol. 26, no. 5, p. 913, 2005.
[30] B. Hosseini, "Dispersion of pollutants in the atmosphere: A numerical study," Ph.D. dissertation, Dept. Math., Simon Fraser Univ., Burnaby, BC, Canada, 2013.
[31] J. M. Stockie, "The mathematics of atmospheric dispersion modeling," *SIAM Rev.*, vol. 53, no. 2, pp. 349–372, Jan. 2011.
[32] S. P. Arya, *Air Pollution Meteorology and Dispersion*, vol. 310. New York, NY, USA: Oxford Univ. Press, 1999.
[33] A. Daly and P. Zannetti, "Air pollution modeling—An overview," in *Ambient Air Pollution*. Half Moon Bay, CA, USA: The Arab School for Science and Technology, 2007, pp. 15–28.

HASSAN SAADAOUI was born in Fkih Ben Salah, Morocco, in 1977. He received the B.S. degree in computer sciences from the Faculty of Sciences Samlalia, Cadi Ayyad University, Marrakech, in 2001, and the M.Sc. degree in telecommunications and computer networks from the Faculty of Sciences, Chouaib Doukkali University, El Jadida, Morroco, in 2005. He has authored several publications in well-known conferences, such as COMMNET, ICSDE, SITA, SADEXS, CMT, and CIFEM. His current research interests include mobile target localization and tracking and distributed control in multiagent system networks.

FAISSAL EL BOUANANI (Senior Member, IEEE) was born in Nador, Morocco, in 1974. He received the M.S. and Ph.D. degrees in network and communication engineering from Mohammed V University–Souissi, Rabat, Morocco, in 2004 and 2009, respectively. He was a Faculty Member with the University of Moulay Ismail, Meknes, from 1997 to 2009. In 2009, he joined the National High School of IT/ENSIAS College of Engineering, Mohammed V University, Rabat, where he is currently a Full Professor. He advised many master's and Ph.D. students at Mohammed V University. He was an Associate Dean of the scientific research and cooperation at the University of Moulay Ismail. So far, his research efforts have culminated in more than 100 papers in a wide variety of IEEE/ACM international conferences and journals. His current research interests include performance analysis and design of wireless communication systems. He has been involved as a TPC member in various conferences and IEEE journals. His Ph.D. thesis was awarded the best one by Mohammed V University–Souissi, in 2010. He served as the TPC Chair for the ICSDE Conferences, the General Co-Chair for ACOSIS 2016 and CommNet 2018 Conferences, and the General Chair for the 2019/2020/2021/20202 CommNet Conferences. He is an Associate Editor of IEEE Access and an Associate Editor of *Frontiers in Communications and Networks* journal. He is also serving as a Lead Guest Editor for the Physical Layer Security Special Issue of *Frontiers in Communications and Networks* journal.