Building Synergetic Consensus for Dynamic Gas-Plume Tracking Applications using UAV Platforms

Athanasios Ch. Kapoutsis[1], Iakovos T. Michailidis[1], Yiannis Boutalis[1] and Elias B. Kosmatopoulos[1]

***Abstract*—This article investigates the problem of deploying a swarm of UAVs equipped with gas sensors for industrial remote gas-plume sensing. This setup's objective is to continuously adjust the swarm formation to maximize the combined perception for the dynamically evolved plume's cloud, focusing around areas with the highest concentration/intensity. Initially, such a setup is formulated into an optimization problem, the solution of which could be acquired by the maximization of an appropriately defined objective function. Due to the model-free approach, this objective function's analytical form is not available, prohibiting standard gradient descent methodologies. To this end, a tracking algorithm is developed and studied, which operates in a distributed manner and enables the UAV swarm to build a common consensus dynamically, during the evolution of the leakage phenomenon. The overall performance is tested in a simulative yet realistic environment using ANSYS Fluent suite, considering a simultaneous gas-leak incident at two different points. Aside from the standalone evaluation study, the proposed gas-plume tracking scheme is able to outperform a state-of-the-art alternative algorithm, namely Efficient Global Optimization (EGO), in various simulation setups, deploying a different number of UAVs on the field.**

***Index Terms*—Remote gas sensing; Swarm Intelligence; Multi-robot; Gas-plume tracking; Autonomous UAVs**

I. Introduction

In certain applications, executing tasks related to inspection and monitoring usually impose practically unbearable financial costs. Inspection and monitoring in remote (difficult for humans to reach) and hazardous environments is one of the most promising application domains for mobile sensory platforms. The Unmanned Aerial Vehicles (UAVs) market has grown significantly within the past years, modernizing daily inspection and remote sensing applications. UAVs are proven capable of gradually or entirely overtaking this responsibility from humans, especially when the risk of human lives is over certain limits i.e., highly hazardous or remote environments. Such platforms' versatility has led to the replacement of humans with such sensory platforms in various domains. Low production costs, satellite communication reliability, as well as increased on-board computational capacity, have rendered the use of unmanned vehicles a mainstream solution for low-complexity operations nowadays. Moreover, the variety of different UAV platforms and sensory add-ons are offering a wide range of new commercial solutions [1] in many industrial domains where faster, safer, easier remote inspection or surveillance are needed: Environmental state monitoring [2]; Powerlines inspection [3]; PV panels inspection [4]; Large buildings structural inspection [5]; Aiding in search & rescue operations [6]; etc.

In recent years, several industrial (e.g., factories, harbors, military hangars, warehouses, etc.) accidents have been caused by either odor or odorless lethal (or flammable) chemical gas leakages putting in direct danger workers, neighboring infrastructure as well as citizens. Several large industrial accidents caused by infrastructure malfunctions and poor maintenance have been reported resulting in devastating losses of human lives. In addition, industrial gas leakages may have other side effects that may severely impact the quality of breathable air or even contribute to greenhouse gases in a long-term manner. Gas emissions from accidental leakages, poorly maintained combustion engines, as well as burnt chemicals, may cause severe respiratory illnesses [7] and environmental problems in high-density urban or suburban areas [8]. Gas emissions also cause phenomena like acid rain, depletion of the ozone layer as well as global warming. Unfortunately, most volatile compounds are usually colorless, tasteless, and odorless and cannot be easily recognized by humans early enough to activate appropriate countermeasures. The list of accidents caused by air-pollutants, poisons, and chemical hazards or radiation is still growing. Among others, one of the most devastating accidents in the history of India; the 1984 Bhopal tragedy [9]; was caused by methyl isocyanate leakage that cost the lives of 3000 people.

Accidents like that have motivated the research community to investigate technological solutions that could effectively identify, recognize, and react rapidly to improve response times and provide valuable insights in time to decision-makers. Currently, gas monitoring systems consist of a static network of sensors that are distributed at the most prominent key locations [10], [11]. However, gas plumes' movement is a dynamic phenomenon affected by several uncertain/unknown factors: leakage source size and intensity, type of leaked gas, surrounding obstacles, wind direction, wind speed, sensor radius/range, etc. Such dynamics can affect the optimal positioning of stationary sensors, hindering the perceived overall situation's quality and certainty. To obtain a representative and always-up-to-date picture of the gas plume spatial distribution, it seems essential to collect spatially distributed concentration measurements from frequently (depending on the phenomenon's inertia) readjusted locations.

[1]Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, Greece. Corresponding author: Athanasios Ch. Kapoutsis (akapouts@ee.duth.gr)

As a result, airborne remote sensing platforms for automated monitoring and mapping applications [2] of dangerous gas leaks and plumes are considered the most appropriate solution to tackle the dynamicity of the moving plume monitoring problem. UAVs have already been demonstrated for civil purposes including various applications of gas detection [12], such as obtaining gas concentration mapping [13], [14], monitoring emissions [15], [16], and gas source localization [17], [18], [19] in environmentally sensitive areas. The most popular approaches in the literature for UAVs swarming are based on biologically-inspired algorithms [20] e.g., Grey Wolf Optimization (GWO) [17], Particle Swarm Optimization (PSO) [21], [22], [23], and Ant Colony Optimization (ACO) [24]. GWO variances are limited by the minimum number of available agents; since all consider dynamically changing roles of the same UAV based on the currently perceived situation e.g., alpha, beta, delta, and omega agents suggest that at least 4 agents should be available [18]. PSO variances are usually significantly affected by the stochasticity of each searcher's velocity that may potentially hinder its performance in highly non-linear problems, requiring several hundreds of iterations to converge [25]. ACO, similar to PSO and GWO, lacks practical applicability in large-scale problems, where several hundreds of iterations are required to converge to the optimal formation after every timestamp; especially in cases where the simulated environment is not static [26], [27] or a model of both the UAVs and the environment is needed for evaluating all candidate configurations [28], [29]. All these factors have not allowed the universal appliance of autonomous swarm systems, and in the vast majority of the real-world scenarios, human operators are still used to operate each UAV manually. To make matters worse, the coordination among all these UAVs' pilots is not easy, especially for time-critical applications.

A. Contributions

Motivated by the aforementioned limitations, a novel approach for multi-UAV swarm path planning and plume monitoring has been developed based on a well-established and thoroughly evaluated [30], [31], [32], [33] model-free optimization methodology – namely Cognitive Adaptive Optimization (CAO) [34], [35]. The distributed plume tracking approach, originally developed in [36], deviates from the aforementioned CAO family of approaches, mainly in its distributed nature. More precisely, at each iteration, each UAV, although it does not have any information regarding i) the decision variables of the other UAVs, ii) their measurements, and iii) the overall accomplishment of mission objectives, is capable of effectively updating its decision variables recognizing its role to the swarm. The latter can be achieved by optimizing a specialized objective function that is tailored to each UAV, which quantifies the contribution of this UAV to the overall mission objectives. Although each UAV optimizes this specialized objective in an isolated way, the swarm can build a common consensus regarding the mission at hand.

In this study, the test scenario simulates, using the powerful ANSYS Fluent suite [37], a realistic case where gas leakage unexpectedly occurs in an indoor industrial environment, and the swarm of UAVs operates in situ where large obstacles and critical infrastructure co-exist. The distributed multi-robot algorithm for plume tracking was able to dynamically adapt to the differentiated dynamics (initial UAV positions, number of available UAVs, etc.) of each experiment and effectively track the simulated gas-leaked plume even from the initial timestamps and maintain/track high-performance levels until the end of the simulation (see section IV for more details); property which is quite crucial especially when critical infrastructure or human lives are in danger and fast response is vital. During the simulated evaluation, the UAVs exhibit a high level of coordination among them: separating tasks, combining locally their sensors' capabilities to adequately capture high-intensity phenomena, etc. It should be highlighted that none of these traits were explicitly designed by a human operator or even included in the objective function. Rather, the proposed approach leaned these behaviors to effectively accomplish the high-level objective. Finally, the proposed gas-plume tracking algorithm's performance is compared against a state-of-the-art algorithm, tailored for optimizing expensive black-box functions, namely Efficient Global Optimization (EGO) [38], [39]. For this evaluation, a pool of 800 different experimental setups were generated. The list consists of 100 experiments with randomly chosen initial positions for each one of the 8 different fleet-sizes, starting from 4 up to 18. The results clearly indicate that the proposed gas-plume tracking scheme outperforms EGO in the achieved value of the objective function for all fleet-sizes, not only at the final step, but also in each timestamp of the plume's evolution. A critical outcome of this evaluation procedure is that for fleet-sizes greater than 12 UAVs, the proposed approach is able to deliver almost the same UAVs' formation, even from the early timestamps, independently of their initial deployment, i.e., negligible standard deviation around mean performance.

B. Paper Structure

The following sections of the paper are organized as: a formulation of the gas-plume tracking problem with respect to the UAVs' positions and the critical objectives considered in the optimization criterion has been included in section II; section III describes the details of a specialized distributed algorithm capable of adequately addressing the previously formulated optimization problem; section IV presents the results and the evaluation outcomes of the considered application; while section V concludes the paper with remarks and the outline of the future works.

II. Problem Formulation

The swarm (team) is consisted of N_r UAVs that can move inside an indoor environment of T_{max} timestamps reflecting the optimization horizon, having as objective to monitor remotely a moving gas-plume caused by spatially-unknown gas-inlet sources.

A. Controllable and State Variables

Without loss of generality, it is assumed that the state vector for each ith UAV, at k timestamp, is defined as

$x_i(k) = [x, y] \in \mathbb{R}^2$, i.e., only the movement in 2D plane is controllable. It assumed that the height for each UAV was fixed to be in-line with the security regulation and meet all the relevant operational constraints (e.g., avoid moving to close to a source or hitting the ceiling).

The considered system evolution dynamics follow a simple yet commonly adopted (e.g, [40]) formula:

$$x_i(k+1) - x_i(k) = u(k+1) \tag{1}$$

where $u_i \in \mathbb{R}^2$ denotes the controllable vector for the ith UAV. The augmented decision state and decision vectors, containing all the UAVs variables, are as follows:

$$x(k) = [x_1(k), x_2(k), \ldots, x_{N_r}(k)] \tag{2}$$

$$u(k) = [u_1(k), u_2(k), \ldots, u_{N_r}(k)] \tag{3}$$

B. Operational Constraints

Controllable vector u should admit feasible values with respect to the problem at hand. In other words, there is a set of constraints that must be met at each timestep. This set of non-linear constraints consists of the following attributes:

1) The UAVs are not allowed to fly over large obstacles (representing very high and/or ceiling mounted industrial machines and infrastructure).
2) The UAVs should remain within the building, avoiding hitting its walls.
3) The UAV infrastructure considers a maximum moving capabilities per timestamp, i.e., $|u_i| \leq u_{max},\ \forall i = 1, 2, \ldots, N_r$.

All the previously defined constraints can be, in their general case, represented as a system of inequalities:

$$\mathcal{C}\left(x(k)\right) \leq 0 \tag{4}$$

where $\mathcal{C}$ includes all the needed inequalities that have to be not positive in order to comply with the aforementioned list of constraints. It must be emphasized, however, that the specified constraints were imposed directly in the dynamics of the system behavior instead of the distributed optimization algorithm (see following section), where feasible solutions were generated complying a priori with the default system dynamics and the simulated operational constraints without requiring feasibility check.

C. Observable Measurements

At each kth timestamp the swarm of UAVs can sense a sub-part of the operational area that lies in the close vicinity of each UAV's location. The exact points of the environment that can be sensed by the ith UAV are defined by its sensing capabilities and the geometry of the local environment, i.e., obstacles' formation. Let $M_i^{(k)}$ to denote the subset of all q points of the environment that can sensed by the ith UAV at kth timestamp. For each point q of the environment the measurements from the ith UAV admit the following form:

$$y_{x_i - q}(k) = \begin{cases} f(k,q) + h_\xi(x_i(k), q)\xi & q \in M_i^{(k)} \\ \textit{undefined} & \text{otherwise} \end{cases} \tag{5}$$

where $f(k, q)$ expresses the true (unknown) concentration of the gas at q point, $h_\xi(x_i(k), q)$ is the multiplicative sensor noise term and ξ is a standard Gaussian noise. The above non-linear sensor model emulates the behavior of "electronic-noses", i.e., collected gas-sensors for e.g., CHx, COx, VOC, NOx gases, as it relates the sensitivity/accuracy of the measurement with the distance from that sensing point. The measurements' vector $y_i(k)$ for the ith UAV at each kth timestamp, is a variable-sized collection of the previously defined measurements:

$$y_i(k) = \left[y_{x_i - q_1}(k), y_{x_i - q_2}(k), \ldots, y_{x_i - q_m}(k)\right], \quad \forall q_j \in M_i^{(k)} \tag{6}$$

where m denotes the current cardinality of $M_i(k)$. The augmented measurements vector is just a concatenation over all UAVs:

$$y(k) = [y_1(k), y_2(k), \ldots, y_{N_r}(k)] \tag{7}$$

D. Objective Function

The monitor formation of the swarm should be dynamically adapted to maximize the combined (accumulated) levels of sensed gas-plume intensity, i.e., synergistically maximize, using the limited sensing capabilities, the coverage of the gas-plume intense areas as its volume expands over time.

The accomplishment of such an objective is quantified in the following instantaneous objective function:

$$\Pi\left(y(k)\right) = \sum_{i=1}^{N_r} \sum_{q \in M_i^{(k)} \cup U} \frac{y_{x_i - q}(k)}{\|x_i - q\|^2} + \sum_{q \in D} \frac{y_{x_{j^*} - q}(k)}{\|x_{j^*} - q\|^2} + \gamma \sum_{q \in M_1^{(k)} \cup M_2^{(k)} \cup \ldots M_{N_r}^{(k)}}, \quad \text{where } j^* = \underset{j, q \in M_i^{(k)}}{\operatorname{argmin}} \|x_j - q\|^2 \tag{8}$$

where U denotes the subset of all points q of the environment sensed *uniquely* by a single UAV, while D contains all the points that have been measured by *at least two* UAVs.

Conceptually, the first two terms of (8) deal with rewarding high-intensity measurements weighted with each measurement's closest distance. More precisely, the second term completely discards all duplicate measurements from other UAVs that have been sensed from a greater distance. By doing so, the system is enabled to develop coordination activities, avoiding wasting the team resources on a single sub-part of the terrain. The third term is responsible for maximizing the overall sensory coverage in the environment, independently of the underlying gas-plume intensity. The idea behind such term is that when the UAVs cannot contribute much to the whole monitoring task (either because the phenomenon is at its early stages or there are plenty of UAVs deployed), they should be as laid out as possible to increase the overall situational awareness of the system. For this term, an extra weight (γ) is added to express the supportive/secondary nature of such an objective. In a nutshell, at each kth timestamp, by maximizing this objective, the swarm is "forced" to utilize wisely its

cumulative sensors' capabilities, spreading the team members over the whole mass of gas while aggregating in areas of high intensity.

Due to the time-variant nature of the problem, the maximum to (8) is not static and changes with respect to the changes in the gas-plume evolution over time. Overall, the mission objective is to acquire the maximum summation of such instantaneous objective instances over the whole incident's horizon, i.e.,

$$\mathcal{J} = \sum_{k=1}^{T_{max}} \Pi\left(y(k)\right) \tag{9}$$

E. Optimization Problem

Based on the aforementioned rationale, the problem of controlling a swarm of UAVs in order to continuously track a dynamic evolving gas-plume, can be equivalently formulated as follows:

$$\begin{aligned} \underset{x(1),x(2),\ldots,x(T_{max})}{\text{maximize}} \quad & \mathcal{J}(x) \\ \text{subject to} \quad & \mathcal{C}\left(x(k)\right) \leq 0,\ k = 1, \ldots, T_{max}. \end{aligned} \tag{10}$$

Due to the time-varying nature of the problem, the maximum of (8) with respect to the swarm configuration in each timestamp is different, or – to be more precise – is gradually changing, rendering it quite challenging for "slow-learning" algorithms. Apart from that, the previous formulated problem has the inherent difficulty that traditional gradient-based methodologies [41] cannot be directly applied, since the analytical formulation for (8) and (4) are not available, as they involve the unknown/unpredictable dynamics of the gas evolution (e.g., the numerators in the first two terms of (8) involve on-the-field measurements). Please note that if we had all the information regarding i) the exact structure of the operational area, ii) the dynamics that govern the UAVs sensors, iii) the precise location of the sources along with their emission rate, iv) the airflow movement over time, and v) the composition of the gas substance, we could calculate these functions analytically. However, to keep our approach as generic as possible, we assume that no information is available a priori. All these features – or at least the ones that matter for the accomplishment of mission objectives (8) – should be learned in real-time during the deployment of the team of UAVs. In addition to the aforementioned obstacles, solving such an extensive system of equations with an increased number of UAVs (that acts as a multiplier to the total number of controllable variables) could be a really burdensome optimization problem.

III. Distributed Gas-Plume Tracking Algorithm

To tackle all these issues, we carefully adapt the general-purpose methodology of [36] to a distributed plume tracking algorithm capable of calculating $x(k)$ on-the-fly, to ultimately optimize the objective function (9). In a nutshell, the algorithm consists of both centralized and distributed operations as described in the following subsections. A flowchart of the core operations inside the distributed gas-plume tracking algorithm is illustrated in figure 1.

A. Centralized Operations

The centralized process of the algorithm focuses on combining the measurements of the different UAVs and automatically assign different –not imposed/designed by any human operator – tasks on each UAV.

• **Step 1** Gather all UAVs' measurements (6) after the execution of $x(k)$ decision variables. It must be emphasized that these steps can be performed even in cases where global communication between all UAVs is not feasible; instead, each robot can send and receive measurements to and from peer UAVs. The latter can be achieved by introducing an additional condition on the constraints set (4), ensuring the connectivity among them.

• **Step 2** Having the set of measurements $y(k)$, we can analytically calculate the objective function using equation (8):

$$\Pi_k = \Pi\left(y(k)\right)$$

• **Step 3** For each UAV calculate the contribution to the accomplishment of the plume tracking objective as follows:

$$\Delta_i(k) = \frac{\Pi_k - \Pi\Big(y_1(k), \ldots, y_{i-1}(k), y_i(k-1), y_{i+1}(k) \ldots, y_{N_r}(k)\Big)}{\|x_i(k) - x_i(k-1)\|} \tag{11}$$

Conceptually, $\Delta_i(k)$ carries information regarding $\frac{\partial \Pi_k}{\partial x_i(k)}$ quantifying the effect of $x_i(k)$ on the current problem instance. Note that $\frac{\partial \Pi_k}{\partial x_i(k)}$ is not available analytically, as the close form that relates $x_i(k)$ with the objective function (8) is environment/dynamics-dependent and generally not available.

• **Step 4** Transmit only each scalar $\Delta_i(k)$ value to the corresponding ith UAV to initiate the distributed optimization that is going to be performed in parallel, without any other exchanging of information.

B. Distributed Operations

During the distributed optimization of the controllable variables of each UAV (1), the on-board controller is called to design the next movement, so as optimize its contribution $\Delta_i(k)$ with respect to the accomplishment of overall mission objectives (8). Each such subproblem is a lower-dimensional optimization problem, and thus can typically be solved more easily and quickly than the full problem.

• **Step 1** Initially the local cost function $J_i(k)$ is updated according to the following formula, where the term "local" refers to the impact of the change in position of ith UAV on the total performance criterion:

$$\mathcal{P}_i(k) = \mathcal{P}_i(k-1) + \Delta_i(k),\ \forall k \geq 2,\ \mathcal{P}_i(1) = \Pi(y(1)) \tag{12}$$

• **Step 2** Fit on the look-up table of the previously acquired data, i.e.,

$$\left[\begin{array}{c|c} x_i(k-W+1) & \mathcal{P}_i(k-W+1) \\ x_i(k-W+2) & \mathcal{P}_i(k-W+2) \\ \vdots & \vdots \\ x_i(k-1) & \mathcal{P}_i(k-1) \\ x_i(k) & \mathcal{P}_i(k) \end{array}\right] \tag{13}$$

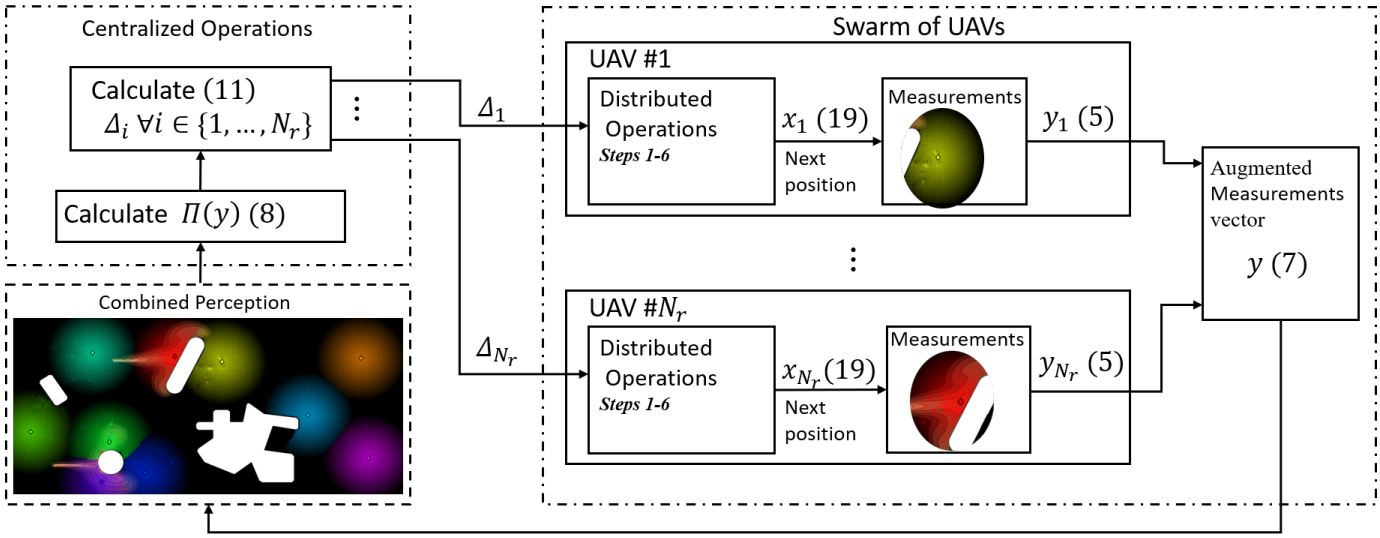


Fig. 1: Flowchart of the distributed tracking algorithm. At each timestep, all the UAVs' measurements y (annotated with the corresponding distance matrix) are compiled to calculate the current value of the objective function $\Pi(y)$. Then, each ith UAV's contribution Δ_i is calculated. This scalar metric Δ_i contains "distilled" information about the progress of the ith UAV with respect to both the plume tracking objectives and the formation of the other UAVs. After transmitting all Δ_i, each UAV calculates –in a fully distributed fashion– its next sensing position, employing the optimization procedure of subsection III-B.

an estimator as follows:

$$\mathcal{P}_i(k) \approx \hat{\mathcal{P}}_i(k) = \theta_i^\tau(k)\phi_i\big(x_i(k)\big) \tag{14}$$

where W denotes the time-window over which the estimation is taking place, ϕ_i is the regression vector and θ_i is the estimator parameters' vector. The estimator vector θ_i is constructed using standard Least-Squares (LS) estimator principles, i.e., θ_i is obtained by solving the following LS optimization problem:

$$\theta_i(k) = \underset{\vartheta}{\operatorname{argmin}} \sum_{\ell=k-W+1}^{k} \Big(\vartheta^\tau \phi_i\big(x_i(\ell)\big) - \mathcal{P}_i(\ell)\Big)^2 \tag{15}$$

$\phi_i(x_i)$ may admit several forms that respect the universal approximation property [42]. For this realization, a set of L 3rd degree monomials is employed.

• **Step 3** Generate randomly a set of B candidate perturbations:

$$\delta x_i^{(1)}(k), \delta x_i^{(2)}(k), \ldots, \delta x_i^{(B)}(k) \tag{16}$$

that respect the operational constraints (4), i.e.,

$$C\left(\left[\bar{X}_1, \ldots, \bar{X}_{i-1}, x_i + \delta x_i^{(b)}, \bar{X}_{i+1}, \ldots, \bar{X}_{N_r}\right]\right) \leq 0, \quad \begin{matrix} \forall b = 1, 2, \ldots, B \\ \forall i = 1, 2, \ldots, N_r \end{matrix} \tag{17}$$

where $\bar{X}_j(k)$ is a 2D sphere with center $x_j(k)$ and radius u_{max} that denotes the possible movements according to (1) for the jth UAV ($j \neq i$), at kth timestamp[1].

The random choice for the candidates is essential and crucial for the efficiency of the algorithm, as such a choice guarantees that $\hat{\mathcal{P}}_i(k)$ is a reliable and accurate estimate for $\mathcal{P}_i(k)$; see [34], [35] for more details.

• **Step 4** Evaluate the set of feasible candidates (16) from previous step on the constructed estimator (14) and pick the candidate perturbation with the "best" effect:

$$b^* = \underset{b=1,\ldots,B}{\operatorname{argmax}} \theta_i^\tau(k)\phi_i\left(x_i(k) + \delta x_i^{(b)}(k)\right) \tag{18}$$

• **Step 5** Set $u_i(k+1) = \delta x_i^{(b^*)}(k)$ utilizing the best candidate and update the next state variables according to (1):

$$x_i(k+1) = x_i(k) + u_i(k+1) \tag{19}$$

• **Step 6** Synchronous update of the current timestamp:

$$k = k + 1$$

Note 1: The above, distributed update of the decision variables does not need information about what is happening to the rest of the UAVs. All the crucial information has been encoded to the scalar value $\Delta_i(k)$. Additionally, it must be emphasized that the calculation of $\Pi\Big(y_1(k), \ldots, y_{i-1}(k), y_i(k-1), y_{i+1}(k) \ldots, y_{N_r}(k)\Big), \forall i = 1, 2, \ldots, N_r$ does not require any new measurements, as the sensor readings that are needed to calculate the global objective function are already available from the previous timestamp.

Note 2: The local convergence of the complete algorithm can be guaranteed in the general case where both unknown function $\Pi(\cdot)$ and each UAV's contribution $\mathcal{P}_i$ are non-convex, non-smooth functions (sketch of proof has been established in [43], [36]), as the behavior of the complete algorithm approximates one of block coordinate descent (BCD) family of approaches, namely ([44], Algorithm 1).

[1]The evaluation of (17) can be performed without the need to notify each UAV about the current swarm configuration, by making the reasonable assumption that each UAV is able to perceive objects – and therefore other UAVs – in a distance that is at least two times its maximum movement capabilities u_{max}.

C. Computational Complexity

During the algorithm's operation, there are two sets of operations that significantly affect the decision time. The first is at the calculation of the objective function (8) during the centralized operations. Having acquired the measurements (6), the complexity for the calculation of the objective function value grows linear with respect to the number of UAVs $\times$ the number of distinct measurements points for each UAV, i.e., $\mathcal{O}(N_r m)$. Considering also (11) the objective function is going to be evaluated N_r times to calculate each $\Delta_i(k)$, the overall complexity of the centralized activities is turned to $\mathcal{O}\left(N_r^2 m\right)$. In general, the calculation of such value is usually very strongly dependent on the underlying sensors' dynamics and the corresponding density in the produced measurements' array. In essence, this is an aspect that forms a trade-off between dense measurements and decision time, that should be considered at the realization of the system. The second pillar of computational complexity is with respect to per UAV calculations. The majority of computational burden for each UAV is accumulated in solving the LS optimization problem of (15). LS optimization problem requires $\mathcal{O}\left(W^2 L + L^3\right)$ [45], where L is number of monomials in each ϕ_i and W the time-window.

IV. Simulation Results and Evaluation

A. Environment Details

To evaluate the performance of the proposed algorithm the high-fidelity ANSYS Fluent Computational Fluid Dynamics (CFD) suite [37] was utilized. The simulation horizon considers 1000 timestamps, each reflecting a 0.3-second time-interval. The simulation environment was defined as a rectangle area of $92 \times 42m^2$ that contains stationary, non-traversable obstacles as depicted in figure 2. For the meshing of the operational area, a standard linear model, specialized for computational fluid dynamics, with a default element size of $5.099m$ is deployed.

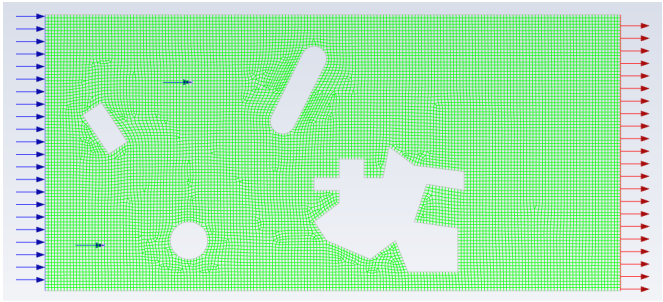
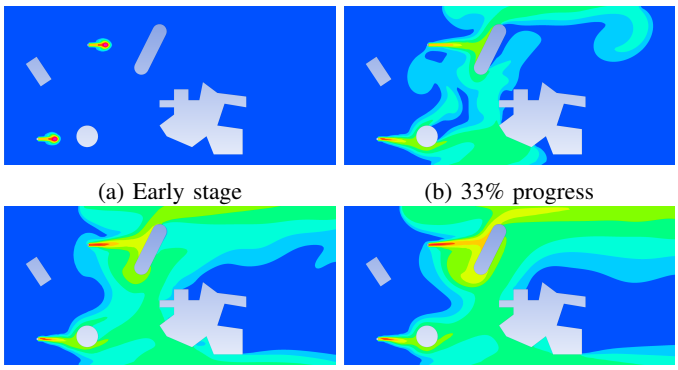


Fig. 2: Experimental 2d plane indoor map. The two blue arrows inside the environment indicate the locations of CH_4 sources. Ventilating air is streamed in from the left side and exhausted to the right-hand side of the environment.

In the adopted scenario, a constant flow of ventilating airstreams with a velocity magnitude of $1m/s$, a turbulent intensity of 10% and hydraulic diameter of $0.44m$ was applied from the left side of the environment as shown in figure 2 which depicts the placement of the two odor sources as well. The chosen substance was methane (CH_4) and was diffused to the environment by two sources that are depicted in figure 2 with blue arrows inside the environment. CH_4 is dispersed with a velocity magnitude of $40m/s$, a turbulent intensity of 10% and a hydraulic diameter of $0.01m$. Figure 3 graphically illustrates the diffusion of CH_4 with aforementioned parameters inside environment of figure 2 for four indicative timestamps.



(a) Early stage (b) 33% progress

(c) 66% progress (d) Final stage

Fig. 3: CH_4 diffusion over the experiment's horizon. The intensity of CH_4 in each point of the environment is represented with a colormap (blue-red).

B. Swarm Realization

Having defined the details of the underlying phenomenon, now the focus is shifted towards the swarm dynamics. It is assumed that all the members of the swarm have homogeneous sensing and moving capabilities. In particular, in all simulation scenarios, all UAVs have a sensing radius of $\rho = 12m$ which also affects their respective sensing reliability, as presented in (5), while all UAVs are capable of repositioning themselves no more than $0.7m$ per timestamp.

The mounted gas sensors are considered to present noisy disturbances as shown in (5), reflecting the sensing reliability as the distance between the sensor and the measured points changes. In specific the multiplicative noise $h_\xi(\cdot)$ of (5) is defined as the squared Euclidean distance between sensor and sensing point, i.e., as the distance between the sensor and the sensing point increases, the metering accuracy/sensitivity decays smoothly; as follows:

$$h_\xi(x_i(k), q) = \|x_i(k) - q\|^2, \ \forall q \in M_i^{(k)}, \forall i = 1, 2, ..., N_r$$

Finally, γ weight in (8) should be set to render the corresponding term auxiliary, but not negligible. Based on the sensors' capabilities and the environment size as defined before, it is found that $\gamma = \rho$ (=maximum sensing radius) is sufficient enough to serve such objective.

C. Algorithm Hyperparameters

Before continuing on the analysis of the results, let's first present the algorithm's utilized values (section III) along with the rational behind each selection. Table I summarizes the free parameters along with the selected values of the distributed

part (subsection III-B). Please note that no free parameters exist on its centralized counterpart (subsection III-A).

TABLE I: Algorithm's parameters (Section III)

Parameter	Value	Equations
Forgetting window	$W = 20$	(13)-(15)
Number of monomials	$L = 10$ randomly chosen 3rd-degree monomials in each $\phi_i(x)$	(15)
Number of candidate perturbations	$B = 100$	(16)

Given the time variant-nature of the problem (8), past tuples $\{x_i(k); \mathcal{P}_i(k)\}$ of (13) quickly become obsolete, and therefore a relatively small window of $W = 20$ is chosen for the LS estimation (15). Extensive analysis in optimization problems with similar set-ups [36], [46], revealed that the utilization of $L = 10$ randomly selected 3rd degree monomials is sufficient enough for 2D control vector space (15). Finally, although, there exist no theoretical results for providing the lower bound of number of random candidate perturbations B (16), several realizations (e.g., [43], [36]) indicate $B > 2 \times$ [the number of controllable parameters], therefore, $B = 100$ is a safe choice for the problem at hand. Overall, and if required, these parameters (Table I) could be manually tuned to achieve improved application-oriented performance.

D. Performance Analysis

To initially assess the performance of the gas-plume tracking scheme (section III) a swarm of $N_r = 10$ UAVs is deployed in random initial positions. Figures 4a - 4d depicts four progress snapshots of the swarm formation with respect to the evolution of CH_4 diffusion. The CH_4 diffusion evolution is illustrated in these figures' background over time, with a colormap, spanning from zero (total black field points) to levels of $\approxeq 0.7$ mass fraction of CH_4 (indicated with light brown). The cyan rhombuses and the corresponding fading cyan region around them represent the UAVs' positions and their field of coverage, respectively.

A close examination of the swarm configuration, especially in figures 4b,4c & 4d, can reveal that all UAVs have been placed strategically over the entire volume of the constantly expanding gas plume; deployed to synergistically track as much as possible of the total intensity of the underlying CH_4 diffusion phenomenon. Moreover, the concentration of UAVs inside the gas-plume was relative to the intensity of the phenomenon, i.e., more UAVs can be observed in the lighter shades of the CH_4 colormap. Please note that, these features were achieved without any global optimization procedure since each local UAV agent did not have explicit access to the actual position of the others nor their measurements, all the information regarding the problem to be solved has been packed inside each scalar $\Delta_i(k)$ (11) that gets transmitted at each kth timestamp to the ith UAV. All the timestamps of this experimental instance have been compiled to a video that can be found under this link[2]

[2]https://youtu.be/9jKD6ORLxgQ

The experimental performance evaluation of the proposed approach is continued with a statistical analysis of the converged swarm formation with respect to the initial deployment. To statistically evaluate the robustness of the approach, the previously described setup with a swarm size of $N_r = 10$ UAVs is repeated 100 times with different random initial positions. Figure 5 aggregates all the positional information for all experiments with the utilization of equipotential lines. As it is expected, equipotential lines in figure 5a have a uniformly random formation, validating the random initial deployment of UAVs. However after some iterations, as it is depicted in 5b, 5c & 5d, the centers of formed clusters of UAVs are in alignment with the intensity of methane. Studying also the result of figure 5 in combination with figure 4, it can be derived that the carefully designed, advantaged positions that were observed in figure 4, were not the outcome of just a simple "good run" nor a result of some favorable initial positions. Actually, the proposed multi-UAV gas-tracking algorithm is consistent in delivering these configurations, independently of the UAVs' initial positions. For demonstration purposes, the evolution of the equipotential lines during the execution of all 100 experiments with random initial deployment of $N_r = 10$ UAVs; has been progressively illustrated in a video under this link[3].

E. Comparison with EGO for Different Swarm Sizes

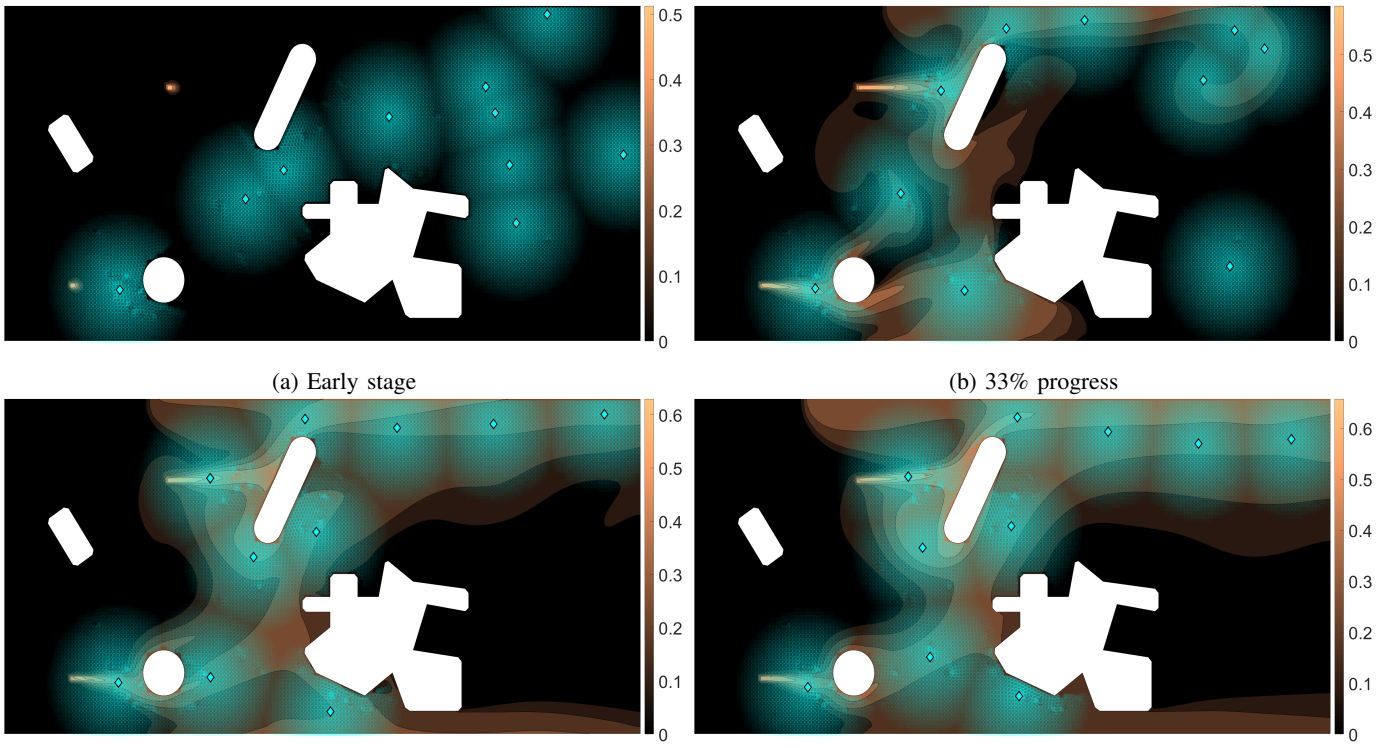
The evaluation analysis is concluded with a comparison study for different number of UAVs on the deployed swarm, between the proposed methodology (section III) and a state-of-the-art methodology for optimizing expensive black-box functions [38], [47], originally described in [39] under the name EGO (Efficient Global Optimization). EGO has been successfully applied in several engineering applications (e.g, [48], [49]), exhibiting efficiency in the optimization of model-free systems, with respect to the total number of actual evaluations on that system. This last remark renders EGO a suitable algorithm for maximizing (8), with a limited number of measurements' vectors (7) that are acquired from applying the decision variables (1) to the actual robotic platforms.

EGO algorithm is employed to optimize the full/centralized problem at hand (10), by directly optimizing the augmented state-space vector (2). The implementation of [47] is utilized (code available from github[4]) with the only modification of forcing the search, in each timestamp, to be subject to the operational constraints (4) of the multi-robot setup.

For this experimentation study, 8 different swarm sizes, i.e., $\{4, 6, 8, 10, 12, 14, 16, 18\}$ UAVs, were employed. For each one of these swarm sizes, 100 initial swarm configurations were randomly generated, forming a pool of 800 different experimental setups. Both the proposed gas-plume tracking approach and EGO algorithm were evaluated on these experimental setups, composing a total of 1600 standalone experiments. Figure 6 summarizes the performance results of such evaluation study. In all eight graphs, the thick colored lines represent the average performance for each timestamp

[3]https://youtu.be/sJeSuCd8ciw

[4]https://github.com/zhandawei/Single_objective_EGO_algorithms



(a) Early stage

(b) 33% progress

(c) 66% progress

(d) Final stage

Fig. 4: Illustrative example of 10 UAVs adjusting their positions to track the time-varying gas-plume phenomenon. Cyan rhombuses and the fading cyan region around them represent the positions of the UAVs and their field of coverage, respectively. The intensity of the underlying CH_4 diffusion is illustrated with the brown-shaded colormap, spanning from zero (black field points) to levels of $\approxeq 0.7$ mass fraction of CH_4 (indicated with light brown tones).

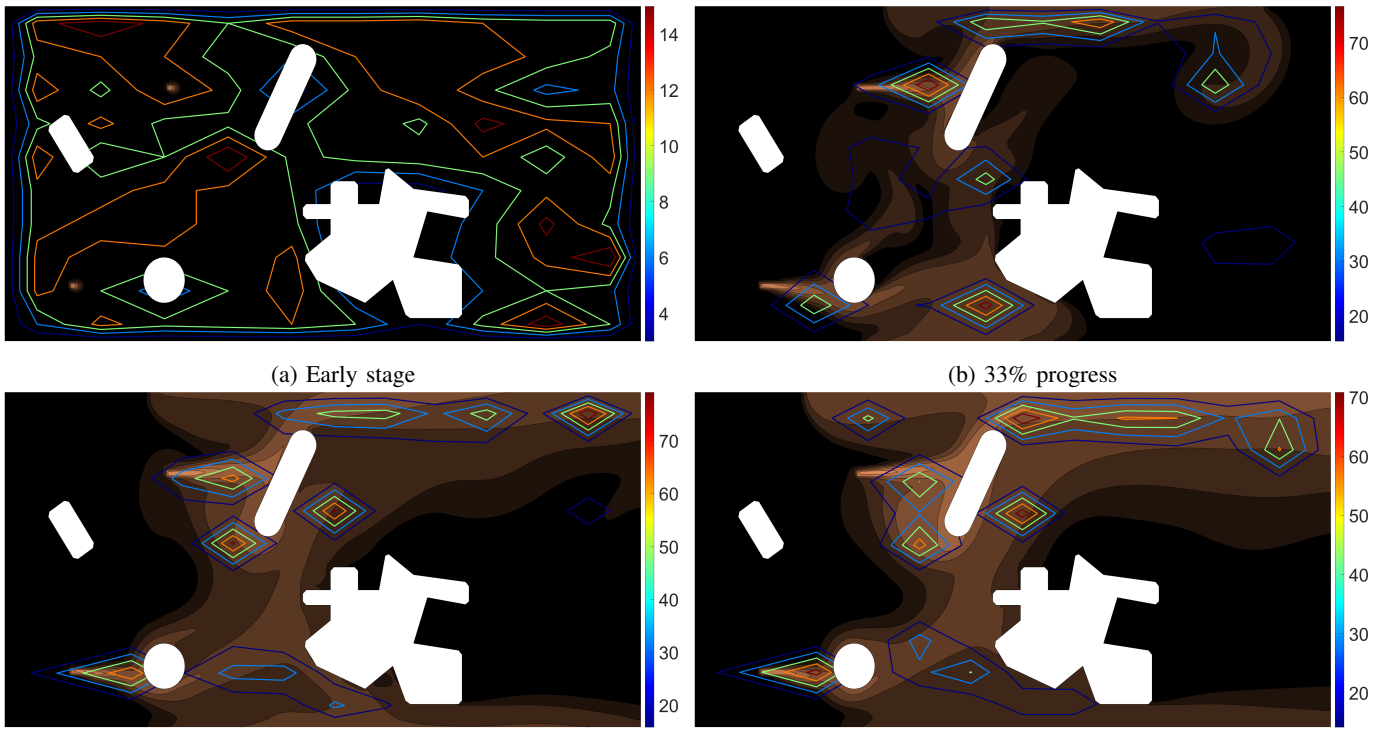
over the set of 100 experiments for the proposed (blue) and EGO (orange) approach. The standard deviation from each average value is depicted with the shaded area around the corresponding thick colored line.

The first remark is that, for all swarm configurations, both algorithms accomplish to improve the overall cognition by constantly improving the swarm formations. Also, as the swarm's size increases, the overall performance value increases too. This is an anticipated behavior, as both decision-making algorithms can spare more UAVs to monitor extensively every part of the plume. Overall, the proposed decision-making scheme outperforms EGO in both average value and standard deviation for all the investigated setups and timestamps. Aside from the average value, the proposed decision-making exhibits extremely low variance, especially in setups with an increased number of UAVs. Technically, for swarm sizes greater than 12 UAVs and if the proposed methodology has been applied, one could be quite confident about the exact formation of UAVs in the environments by just knowing the current timestamp. Indicatively, the average (over all different swarm sizes) standard deviation for the final timestamp is 6.8 and 42.63 for the proposed decision-making scheme and EGO, respectively.

V. Conclusions and Future Work

A semi-distributed algorithm for plume tracking using a team of UAVs has been proposed. Initially, the problem has been transformed into a constrained optimization problem, proposing an objective function scheme to be optimized with respect to swarm configuration. However, the optimization of such an objective function cannot be performed using standard gradient descent methodologies; therefore, a specialized optimization scheme is developed to approximate the solution of such an optimization problem. One of the fundamental elements of this proposed algorithm is that it is not specifically tailored to the dynamics of either UAVs or the environment; instead, it learns, from the real-time sensor measurements, exactly the most effective formations of the swarm for the underlying plume-tracking task. A simulator that utilizes the high-fidelity dynamics of ANSYS Fluent suite was developed to evaluate the proposed scheme's performance. The results of such evaluation proved that the plume-tracking algorithm can optimize the swarm configuration in real-time, adapting to the number of operational UAVs, independent of their initial positions. A cornerstone of the experimental evaluation is the side-by-side evaluation with EGO, an algorithm tailored for optimizing expensive black-box functions. In the reported analysis, the monitor positions calculated from the proposed gas-plume tracking algorithm were consistently improved compared with the EGO case, having also low dependence on the initial deployment.

Several avenues of exploration are left open for future work. One direction is to develop a real-life testbed where we could apply the proposed gas tracking algorithm. Apart from the



(a) Early stage

(b) 33% progress

(c) 66% progress

(d) Final stage

Fig. 5: Statistical equipotential lines aggregating 100 experiments with different initial swarm formation for the case of $N_r = 10$ UAVs over the experiments horizon. The colorbar on the right side of each figure depicts the mapping between color-lines and number of robots found on the corresponding areas.

swarm of UAVs with the appropriate sensor capabilities, one should design a safe-n-secure operational area for the UAVs to perform the tasks autonomously. Strong candidates for such testbeds are abandoned factories and other vast – empty of people – structures that allow the swarm of UAVs to operate freely without any time constraint or other limitation. Another direction could be the appliance in a similar domain, e.g. oil spill tracking, to check the performance of the proposed multi-robot tracking scheme under the sea-related dynamics.

VI. Acknowledgments

This research is carried out / funded in the context of the project "Development and evaluation of an optimal decision-making algorithm for cooperative autonomous vehicles" (MIS 5050057) under the call for proposals "Supporting researchers with emphasis on young researchers - cycle B" (EDULLL 103). The project is co-financed by Greece and the European Union (European Social Fund- ESF) by the Operational Programme Human Resources Development, Education and Lifelong Learning 2014-2020.

References

[1] N. Mohamed, J. Al-Jaroodi, I. Jawhar, A. Idries, and F. Mohammed, "Unmanned aerial vehicles applications in future smart cities," *Technological Forecasting and Social Change*, vol. 153, p. 119293, 2020.

[2] M. Rossi, D. Brunelli, A. Adami, L. Lorenzelli, F. Menna, and F. Remondino, "Gas-drone: Portable gas sensing system on uavs for gas leakage localization," in *SENSORS, 2014 IEEE*. IEEE, 2014, pp. 1431–1434.

[3] D. Jones, "Power line inspection-a uav concept," in *2005 The IEE Forum on Autonomous Systems (Ref. No. 2005/11271)*. IET, 2005, pp. 8–pp.

[4] P. B. Quater, F. Grimaccia, S. Leva, M. Mussetta, and M. Aghaei, "Light unmanned aerial vehicles (uavs) for cooperative inspection of pv plants," *IEEE Journal of Photovoltaics*, vol. 4, no. 4, pp. 1107–1113, 2014.

[5] H. Yoon, J. Shin, and B. F. Spencer Jr, "Structural displacement measurement using an unmanned aerial system," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 3, pp. 183–192, 2018.

[6] E. Lygouras, N. Santavas, A. Taitzoglou, K. Tarchanidis, A. Mitropoulos, and A. Gasteratos, "Unsupervised human detection with an embedded vision system on a fully autonomous uav for search and rescue operations," *Sensors*, vol. 19, no. 16, p. 3542, 2019.

[7] F. M. Henretig, M. A. Kirk, and C. A. McKay Jr, "Hazardous chemical emergencies and poisonings," *New England journal of medicine*, vol. 380, no. 17, pp. 1638–1655, 2019.

[8] E. Bakogiannis, C. Kyriakidis, M. Siti, and V. Eleftheriou, "Four stories for sustainable mobility in greece," *Transportation Research Procedia*, vol. 24, pp. 345–353, 2017.

[9] R. Varma and D. R. Varma, "The bhopal disaster of 1984," *Bulletin of Science, Technology & Society*, vol. 25, no. 1, pp. 37–45, 2005.

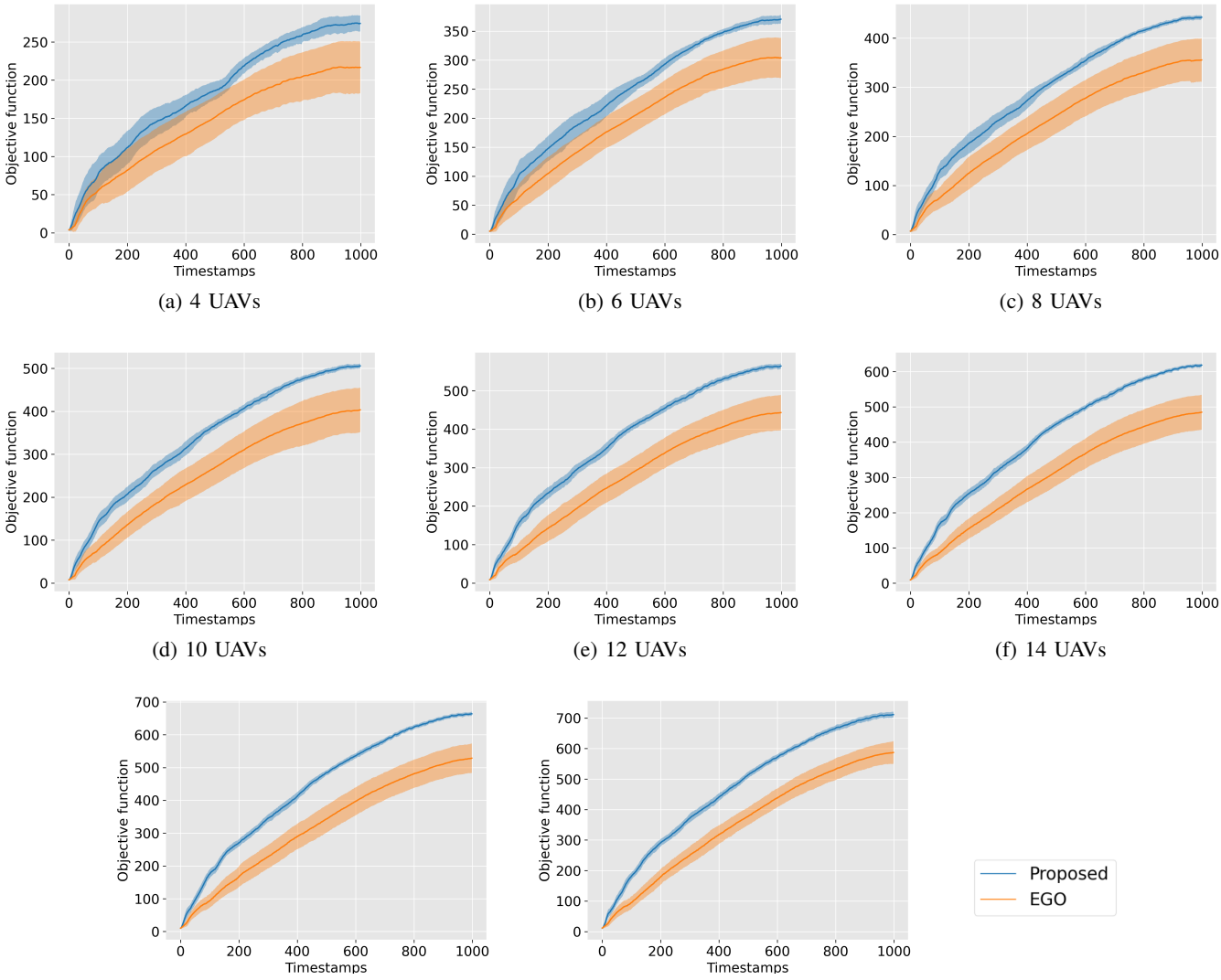
[10] W. Tsujita, A. Yoshino, H. Ishida, and T. Moriizumi, "Gas sensor network for air-pollution monitoring," *Sensors and Actuators B: Chemical*, vol. 110, no. 2, pp. 304–311, 2005.

[11] W. Tsujita, S. Kaneko, T. Ueda, H. Ishida, and T. Moriizumi, "Sensor-based air-pollution measurement system for environmental monitoring network," in *TRANSDUCERS'03. 12th International Conference on Solid-State Sensors, Actuators and Microsystems. Digest of Technical Papers (Cat. No. 03TH8664)*, vol. 1. IEEE, 2003, pp. 544–547.

[12] J. Burgués and S. Marco, "Environmental chemical sensing using small drones: A review," *Science of The Total Environment*, p. 141172, 2020.

[13] J. Burgués, V. Hernández, A. J. Lilienthal, and S. Marco, "Smelling nano aerial vehicle for gas source localization and mapping," *Sensors*, vol. 19, no. 3, p. 478, 2019.

[14] P. P. Neumann, S. Asadi, A. J. Lilienthal, M. Bartholmai, and J. H. Schiller, "Autonomous gas-sensitive microdrone: Wind vector estimation and gas distribution mapping," *IEEE robotics & automation magazine*, vol. 19, no. 1, pp. 50–61, 2012.



(a) 4 UAVs (b) 6 UAVs (c) 8 UAVs

(d) 10 UAVs (e) 12 UAVs (f) 14 UAVs

(g) 16 UAVs (h) 18 UAVs

Fig. 6: Side-by-side comparison, for different number of UAVs in the deployed swarm, between the proposed gas-plume tracking scheme and EGO methodology.

[15] K. Daniel, B. Dusza, A. Lewandowski, and C. Wietfeld, "Airshield: A system-of-systems muav remote sensing architecture for disaster response," in *2009 3rd Annual IEEE Systems Conference*. IEEE, 2009, pp. 196–200.

[16] D. Caltabiano, G. Muscato, A. Orlando, C. Federico, G. Giudice, and S. Guerrieri, "Architecture of a uav for volcanic gas sampling," in *2005 IEEE Conference on Emerging Technologies and Factory Automation*, vol. 1. IEEE, 2005, pp. 6–pp.

[17] S. Mamduh, K. Kamarudin, A. Shakaff, A. Zakaria, R. Visvanathan, A. Yeon, L. Kamarudin, and A. Nasir, "Gas source localization using grey wolf optimizer," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 1-13, pp. 95–98, 2018.

[18] D. Facinelli, M. Larcher, D. Brunelli, and D. Fontanelli, "Cooperative uavs gas monitoring using distributed consensus," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1. IEEE, 2019, pp. 463–468.

[19] V. Gallego, M. Rossi, and D. Brunelli, "Unmanned aerial gas leakage localization and mapping using microdrones," in *2015 IEEE Sensors Applications Symposium (SAS)*. IEEE, 2015, pp. 1–6.

[20] A. L. Nelson, G. J. Barlow, and L. Doitsidis, "Fitness functions in evolutionary robotics: A survey and analysis," *Robotics and Autonomous Systems*, vol. 57, no. 4, pp. 345–370, 2009.

[21] J. Sánchez-García, D. Reina, and S. Toral, "A distributed pso-based exploration algorithm for a uav network assisting a disaster scenario," *Future Generation Computer Systems*, vol. 90, pp. 129–148, 2019.

[22] G. Ferri, E. Caselli, V. Mattoli, A. Mondini, B. Mazzolai, and P. Dario, "Explorative particle swarm optimization method for gas/odor source localization in an indoor environment with no strong airflow," in *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2007, pp. 841–846.

[23] L. Marques, U. Nunes, and A. T. de Almeida, "Particle swarm-based olfactory guided search," *Autonomous Robots*, vol. 20, no. 3, pp. 277–287, 2006.

[24] Y. Zou, D. Luo, and W. Chen, "Swarm robotic odor source localization using ant colony algorithm," in *2009 IEEE International Conference on Control and Automation*. IEEE, 2009, pp. 792–796.

[25] D. E. Gbenga and E. I. Ramlan, "Understanding the limitations of particle swarm algorithm for dynamic optimization tasks: A survey towards the singularity of pso for swarm robotic applications," *ACM Computing Surveys (CSUR)*, vol. 49, no. 1, pp. 1–25, 2016.

[26] Z. Abdmouleh, A. Gastli, L. Ben-Brahim, M. Haouari, and N. A. Al-Emadi, "Review of optimization techniques applied for the integration of distributed generation from renewable energy sources," *Renewable Energy*, vol. 113, pp. 266–280, 2017.

[27] W. J. Gutjahr, "Convergence analysis of metaheuristics," in *Matheuristics*. Springer, 2009, pp. 159–187.

[28] G. Calabrò, V. Torrisi, G. Inturri, and M. Ignaccolo, "Improving inbound logistic planning for large-scale real-world routing problems: a novel ant-colony simulation-based optimization," *European Transport Research Review*, vol. 12, pp. 1–11, 2020.

[29] S. Perez-Carabaza, E. Besada-Portas, J. A. Lopez-Orozco, and M. Jesus, "Ant colony optimization for multi-uav minimum time search in uncertain domains," *Applied Soft Computing*, vol. 62, pp. 789–806, 2018.

[30] S. Baldi, I. Michailidis, H. Jula, E. B. Kosmatopoulos, and P. A. Ioannou, "A "plug-n-play" computationally efficient approach for control design of large-scale nonlinear systems using co-simulation," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 436–441.

[31] S. Baldi, I. Michailidis, V. Ntampasi, E. B. Kosmatopoulos, I. Papamichail, and M. Papageorgiou, "Simulation-based synthesis for approximately optimal urban traffic light management," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 868–873.

[32] A. C. Kapoutsis, S. A. Chatzichristofis, L. Doitsidis, J. B. de Sousa, and E. B. Kosmatopoulos, "Autonomous navigation of teams of unmanned aerial or underwater vehicles for exploration of unknown static & dynamic environments," in *21st Mediterranean Conference on Control and Automation*. IEEE, 2013, pp. 1181–1188.

[33] I. T. Michailidis, D. Manolis, P. Michailidis, C. Diakaki, and E. B. Kosmatopoulos, "Autonomous self-regulating intersections in large-scale urban traffic networks: a chania city case study," in *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2018, pp. 853–858.

[34] E. B. Kosmatopoulos and A. Kouvelas, "Large scale nonlinear control system fine-tuning through learning," *IEEE Transactions on Neural Networks*, vol. 20, no. 6, pp. 1009–1023, 2009.

[35] E. B. Kosmatopoulos, "An adaptive optimization scheme with satisfactory transient performance," *Automatica*, vol. 45, no. 3, pp. 716–723, 2009.

[36] A. C. Kapoutsis, S. A. Chatzichristofis, and E. B. Kosmatopoulos, "A distributed, plug-n-play algorithm for multi-robot applications with a priori non-computable objective functions," *The International Journal of Robotics Research*, vol. 38, no. 7, pp. 813–832, 2019.

[37] I. ANSYS, "Ansys fluent user's guide, release 19.0," *Equation*, no. 6.68, 2018.

[38] S. Daulton, M. Balandat, and E. Bakshy, "Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization," *arXiv preprint arXiv:2006.05078*, 2020.

[39] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[40] S. S. Mansouri, C. Kanellakis, B. Lindqvist, F. Pourkamali-Anaraki, A.-A. Agha-Mohammadi, J. Burdick, and G. Nikolakopoulos, "A unified nmpc scheme for mavs navigation with 3d collision avoidance under position uncertainty," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5740–5747, 2020.

[41] Y. Nesterov, "Gradient methods for minimizing composite functions," *Mathematical Programming*, vol. 140, no. 1, pp. 125–161, 2013.

[42] M. M. Polycarpou and P. A. Ioannou, *Identification and control of nonlinear systems using neural network models: Design and stability analysis*. Citeseer, 1991.

[43] D. I. Koutras, A. C. Kapoutsis, and E. B. Kosmatopoulos, "Autonomous and cooperative design of the monitor positions for a team of uavs to maximize the quantity and quality of detected objects," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4986–4993, 2020.

[44] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, 2015.

[45] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013, vol. 3.

[46] A. C. Kapoutsis, S. A. Chatzichristofis, L. Doitsidis, J. B. de Sousa, J. Pinto, J. Braga, and E. B. Kosmatopoulos, "Real-time adaptive multi-robot exploration with application to underwater map construction," *Autonomous robots*, vol. 40, no. 6, pp. 987–1015, 2016.

[47] D. Zhan, J. Qian, and Y. Cheng, "Pseudo expected improvement criterion for parallel ego algorithm," *Journal of Global Optimization*, vol. 68, no. 3, pp. 641–662, 2017.

[48] J. P. Janet, S. Ramesh, C. Duan, and H. J. Kulik, "Accurate multiobjective design in a space of millions of transition metal complexes with neural-network-driven efficient global optimization," *ACS central science*, vol. 6, no. 4, pp. 513–524, 2020.

[49] D. Cinquegrana and E. Iuliano, "Efficient global optimization of a transonic wing with geometric data reduction," in *35th AIAA Applied Aerodynamics Conference*, 2017, p. 3057.