IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 13, 2021, accepted September 28, 2021, date of publication October 1, 2021, date of current version October 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3116919

Information Sharing Based on Local PSO for UAVs Cooperative Search of Moved Targets

HASSAN SAADAOUI[1], FAISSAL EL BOUANANI[1], (Senior Member, IEEE), AND ELMEHDI ILLI[2], (Member, IEEE)
[1]ENSIAS College of Engineering, Mohammed V University of Rabat, Rabat 10000, Morocco
[2]Center for Cyber-Physical Systems, Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates

Corresponding author: Faissal El Bouanani (f.elbouanani@um5s.net.ma)

ABSTRACT This paper proposes an optimization strategy for searching moving targets' locations using cooperative unmanned aerial vehicles (UAVs) in an unknown environment. Such a strategy aims at reducing the overall search time and impact of uncertainties caused by the motion of targets, as well as improving the detection efficiency of UAVs. Specifically, we report, based on the UAV's scan of a location and taking into account (i) the detection and communication coverage limitations, and (ii) either a false alarm or inaccurate detection of the target, either the existence or the absence of the target. Moreover, leveraging a cooperative and competitive particle swarm optimization (PSO) algorithm, a decentralized target search model, relying on a real-time dynamic construction of cooperative UAV local sub-swarms (LoPSO), is proposed. Each sub-swarm strives to validate quickly the target location, updated based on the Bayesian theory. In such a strategy, each UAV operates in two flight modes, namely, either in swarm mode or in Greedy mode, and takes into consideration the received data from other UAVs to improve the overall environmental information. The simulation results revealed that the LoPSO outperforms other well-known searching methods of target methods for target search in unknown environments in terms of both performance and computational complexity.

INDEX TERMS Cooperative search, decision making, moved target, particle swarm optimization, unmanned aerial vehicle (UAV).

I. INTRODUCTION

The cooperative control of significant small unmanned aerial vehicles (UAVs) groups is less expensive and of great interest in military and civilian domains, such as space exploration, forest fire watch, patrols, and search and rescue. These tasks can be critical, making them ideal for autonomous vehicles [1], [2]. In these types of applications, the UAV group must have numerous capabilities including aircraft communication, navigation, and collision avoidance [3]. Part of the growing interest in UAVs stems from the growing feasibility of these goals due to the improvement in small processors, sensors, cameras, and reliable wireless networks. The cooperative control of multiple UAVs is one of the interesting and key topics in the area of UAV applications. In this regard, different models were proposed, especially the cooperative search target, such as Markov decision-making process, which considers the uncertainty relying on the environmental conditions [4], mathematical planning [5], heuristic algorithms [6], [7], market-based approaches [8], [9], reinforcement learning [10], [11], minimum duration search algorithms (MTS) [12], based on search trajectories minimizing target detection time, random search (RS), in which each drone operates independently of other UAVs, and Greedy methods [13]. These algorithms can mainly be classified into three categories, namely, (i) centralized, (ii) decentralized [14], and (iii) hybrid approach. While the first approach lies in centralizing all swarm search data on a centralized node, the second one relies on sharing the neighborhoods information between swarm UAVs, making the overall system scalable and more robust to a single node data loss. Nevertheless, the centralized approach can achieve the global optimal solution with the cost of central node burdening. In contrast, the hybrid approach [15] uses both decentralized and centralized architectures that typically assume the formation of sub-swarms, leading to the communication flow reduction. Indeed, due to reduced coverage, the internal sub-swarm communication is

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

limited and feasible only between UAVs of the same sub-swarm and is assumed to be available only if the information sharing is judged mandatory.

The research works proposed in the field of multi-UAVs cooperation are of greatest interest in the target search field as the use of UAVs offer several advantages over manned missions, such as reducing the resources use and subsequently the overall operation cost, as well as, obtaining solutions with significantly low computational complexity and high performance. On the other hand, the particle swarm optimization (PSO) algorithm, referred to as a population-based stochastic technique, is mainly inspired by the social behavior of bird flocking to solve practical optimization problems [16]. The strengths of this algorithm are numerous. Foremost, contrary to other concurrent algorithms, PSO has very few parameters to be adjusted, while the computational complexity is significantly reduced. Also, the global optimum solution is most likely achieved [17].

Relying on the PSO algorithm, we propose an efficient hybrid approach based on a local sub-swarm structure (LoPSO) [18] to exchange and update efficiently the search information to reduce the overall search time for finding all missed targets. Pointedly, such information is updated with the aid of target existence probabilities provided by the UAVs, as well as a mapping of the uncertainty. However, the higher the probability value, the higher the probability of the target's existence.

A. RELATED WORK

At present, a large number of works have been realized for efficient and practical UAVs for searching and tracking targets. However, with the significant computational complexity of search problems, in particular, the number of missed targets, their movement behavior, absence of sufficient information on the environment, and especially, the time criticality, it is extremely difficult, if not impossible, that a UAV performs individually the search missions. These latter require the coordination of several UAVs, referred to as multiple UAVs cooperative search [19]. Various approaches dealing with the search optimization problem have been proposed. In [11], Markov Decision Process (MDP) with the use of deep end learning is used, while in [20], a distributed model predictive control is presented, therein a centralized online optimization problem is divided into a decentralized optimization problem by a task-assignment to each drone. Likewise, the K-shortest path routing algorithm was introduced to guide drones to explore static targets in unknown areas [21]. Also, a distributed tasks integrating dynamic programming-based approach, aiming to obtain all necessary information from the swarm of UAVs is proposed in [22]. Furthermore, each UAV is able to determine its optimal flight route. Although these methods are efficient for cooperative UAVs' search problems, they are limited to static targets. In recent years, extensive researches have been devoted to the exploration of moving targets. For instance, emerging cooperative decision and control algorithms based on the aforementioned approaches have been proposed, such as custom combinatorial optimization methods and nature-inspired techniques. These last has become more prevalent due to its effectiveness in dealing with UAV dynamic constraints and the capacity to search for the global optimum in complex scenarios. A variety of nature-inspired algorithms have been developed for UAV search targets such the genetic algorithm (GA) [7], ant colony algorithm (ACO) [23], and particle swarm optimization (PSO) [16], [24]. Among them, the PSO method is one of the most prominent optimization algorithms, and it has been utilized in many applications [25].

In [26], [27], the authors proposed a distributed particle swarm optimization (DPSO) method based on creating a decentralized multi-target tracking system with limited sensor capability to track numerous targets. The whole swarm is split into numerous subgroups, with each subgroup focusing on a single target. In [28], a proposed motion-encoded PSO algorithm is applied to the UAV's search for moving targets by encoding the search trajectory as a series of UAV motion paths evolving over the generation of particles in a PSO algorithm. In [29], an Extended PSO (EPSO) method is proposed that incorporates adaptive inertia weights and obstacle avoidance variables while taking into account the limitations of restricted communication. It proposes utilizing a non-holonomic model to update the forward speed and rotational velocity of the robot swarm in order to accomplish motion control.

Although the aforesaid algorithms have enabled multi-UAV cooperative search, they would face various challenges such as the absence of an efficient cooperation structure, and mainly a lack of an efficient process for the environmental information exchange. To remedy the situation, algorithms based on swarm intelligence have been rolled out. In fact, a number of autonomous UAV can be automatically and adaptively coordinated to achieve a search mission in an unknown area, by setting up an information-sharing mechanism of a local map of the search area (search card) [13], [30] which serves as the UAV's knowledge base on the state of the search area. In contrast, during the exploration, each UAV observes a part of the search area, detects information using onboard surveillance sensors (e.g., cameras), updates its search maps, and shares it with its neighboring UAVs. Afterward, the UAVs must therefore decide the next motion direction, the appropriate timing for such a movement, along with the useful information to exchange periodically.

The cooperative drone search process can be then outlined as follows [31]:

- Sensing the search region and updating the set of probabilities with the aid of UAVs,
- Making decisions about the next search area based on the available information and the upcoming flight mode,
- Forming the sub-swarms for simultaneous exploration of the area of search,
- Sharing the collected information between the sub-swarms.

B. CONTRIBUTION

Motivated by the foregoing, we present an efficient approach based on a dynamic cluster structure that combines the characteristics of a distributed and centralized approach, with the goal of moving the cooperative and controlled autonomously UAVs to undeveloped areas, as well as finding and tracking moving targets in restricted areas. To create a controlled scenario quite close to practical situations, certain assumptions were made on the dynamic behavior of the targets, for the sake of simplicity. Besides, we assume that the targets' speeds vary progressively and follow regular movement directions. This means that they will not face huge fluctuations suddenly. Subsequently, the target motion direction is changed progressively as well. Moreover, the quadrotor UAVs with rotary wings, suitable for operations requiring maneuverability, tight turns, and back-and-forth movements are considered. Of note, the motion control, image processing, low-level control of the actuators, and the types of sensors are beyond the scope of this article. To this end, it is assumed that the UAVs' sensors recognize the visible targets.

In this paper, a local PSO (LoPSO) strengthened algorithm cooperative search for distributed unmanned aerial vehicles (UAVs) is proposed. The whole swarm is divided into sub-swarms in accordance with the communication range depending on the search state. Each sub-swarm ensures the exchange only between its UAVs, while specific UAVs are dedicated to playing the relay role between such a sub-swarm and the base station (BS). Further, the sub-swarms validate periodically the target found by one of their UAVs.

Specifically, a hybrid approach, assuming the formation of sub-swarms using local communication networks to share various useful information is presented. To this end, our main contribution can be summarized as follows:

- We present a new strategy relying on LoPSO with the divided and conquer method,
- Unlike the simple assumption made in [18], all the targets are supposed in motion,
- To gain more insights into the performance and the computational complexity, we demonstrate that our proposed algorithm outperforms the robust well-known similar algorithms.

C. ORGANIZATION OF THE PAPER

The rest of this paper can be structured as follows: Section II describes both system parameters and constraints, while Section III outlines the dynamic model of the UAVs. Section IV presents the proposed LoPSO algorithm, whereas the simulation results alongside useful discussions are provided in Section V. Finally, a conclusion summarizing the outcomes of this work and pointing out some future directions is given in Section VI.

II. SYSTEM MODEL

As shown in Fig.1, we consider a set of N_T moving targets $\{T_i\}_{1\leqslant i\leqslant N_T}$, e.g., a lost person, source of fire in the forest, etc, in a predefined search region Ω, supposed rectangular and split into $L_x \times L_y$ equal-area square cells of length h i.e., its surface equals h^2, where L_x and L_y denote the number of rows and columns, respectively. Furthermore, a set of N_u UAVs $\{U_i\}_{1\leqslant i\leqslant N_u}$ flying over the considered region are cooperatively tracking the aforesaid targets.

TABLE 1. Potential values of the message *I*.

n	message I to send/receive
0	True, False, Neighbors, and Free
1	$P^{(i)}(t)$
2	$C^{(i)}(t)$
3	The target position: $\chi_i^{(t)}$
4	$\mathcal{T}$

For the ease of exposition, let us introduce the following notations:

- N_u: Number of UAVs,
- N_T: Number of missed targets,
- $\mathcal{T}$: Set of found targets,
- $\mathcal{R}$: Communication range of a UAV,
- N_c: Number of cells covered by an UAV, i.e. $N_c = \mathcal{R}/h$,
- t_k: Instance time of validating the kth target,
- $D_{\max}$: Maximum search duration,
- D_{PSO}: Maximum PSO duration,
- τ: Sensing period,
- N_t: Number of targets,
- θ: Probability threshold of a target confirmation,
- E_t: Minimum Energy threshold to stop searching,
- E_i: Current Energy of U_i,
- F_i: Flight mode of U_i,
- $\mathcal{N}_t(U_i)$: Set of indices referring to the neighbors of U_i within $\mathcal{R}$ at a given time t,
- $|\mathcal{N}_t(U_i)|$: Number of elements of $\mathcal{N}_t(U_i)$,
- $\chi_i^{(t)}$: Position of ith particle,
- $M_i(n, I)$: Spreading the message I by either U_i when $i \geq 1$ or the BS if $i = 0$, where n defines the sent information type, whereas I potential values are given in Table 1. Especially, when $n = 0$, the types of messages can be summarized to 4, namely (i) True: indicate that the search is still in progress, (ii) False: indicate to stop the search, (iii) Neighbors: request to join a sub-swarm, and (iv) Free: report to members of a sub-swarm to leave the corresponding sub-swarm.
- $R_i(n, I)$: Receiving the message I from either U_i when $i \geq 1$ or the BS if $i = 0$, where n is an integer defines the received information type. Note that $R_i(n, I)$ is triggered if a transmission is detected,
- $S_i(x, y)$: Sensing the cell $c_{x,y}$ by U_i. It returns the observation in this cell at a given time, i.e., equals 0 or 1 in the absence or presence of a target observation, respectively.

Let $c_{x,y}$ denotes the coordinate of a cell center, where $x \in \{1, .., L_x\}$ and $y \in \left\{1, .., L_y\right\}$. Each cell can store some useful information, such as the uncertainty degree of the environment and target existence probability. In what follows, the cell centered by $c_{x,y}$ is referred by its center for the sake of brevity. Also, we assume that all targets remain in the search

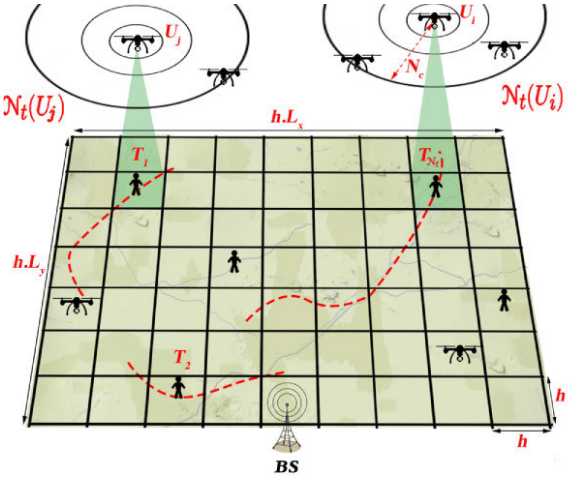


FIGURE 1. Illustrative example of cooperative UAVs tracking moving targets.

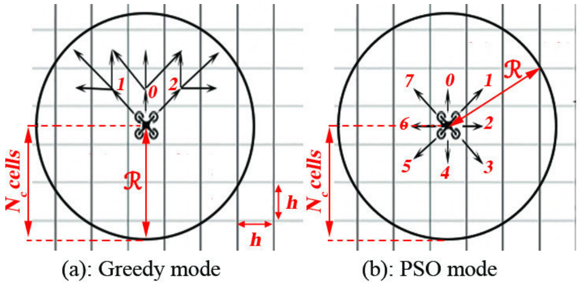
area Ω and that their location information sensed by U_i at the time step t are modeled with a Bernoulli-distribution [30] as

$$P_{x,y}^{(i,t)} = \Pr\left(X_{x,y}^{(i,t)} = 1\right), \tag{1}$$

with $X_{x,y}^{(i,t)}$ denoting the random variable describing the detection of a target in the cell $c_{x,y}$ by U_i, with $X_{x,y}^{(i,t)} = 1$ if the target exists and $X_{x,y}^{(i,t)} = 0$ if not. Initially, as no information is provided on the target existence, we set $P_{x,y}^{(i,0)} = 0.5$ for all cells and UAVs.

By contrast, U_i is modeled as a nonholonomic point mass moving synchronously in the discrete-time with a minimum turning radius, at a slightly different altitude above the search region so as to avoid potential collisions with each other. In addition, U_i makes periodically, at each multiple of the period τ, i.e., $t = m\tau$ where m is a positive number, the decisions by adjusting its current position, and either moves to the next cell, according to its own flying experience and those of the neighbors or stay unmoved. For the sake of simplicity, it is assumed that U_i flies with speed V_i in a specific direction and operates in two main modes:

- *Greedy* mode: In this mode, the UAV moves towards the cells with a minimum number of visits and whose existence probability value of the target is higher. Further, each UAV explores individually as much as possible the entire search area, which imposes the optimization of energy consumption. Towards this end, only three future directions are allowed to the UAV, namely, left, straight, or right with a turning angle of 45°, as depicted in Fig. 2a, corresponding to three different future positions, i.e., $\left\{c_{x_j,y_j}\right\}_{j=0..2}$.
- *PSO mode*: Under this mode, the UAV flies in a swarm which serves to cooperatively validate the position of a target in a short search time compared to the *Greedy* mode. Therefore, further flight future directions, specifically, eight possible directions, as shown in Fig. 2b, i.e., north, northeast, east, southeast, south, southwest, west, and northwest formulated by the centers $\left\{c_{x_j,y_j}\right\}_{j=0..7}$ are authorized in this mode. This allows to balance the UAV's energy consumption, considered significantly smaller than its *Greedy* counterpart.



(a): Greedy mode (b): PSO mode

FIGURE 2. Future directions of the UAV per mode for $N_c = 3$.

Regardless of the aforesaid modes, each UAV U_i has sufficient energy autonomy to remain inside Ω and carry out the necessary operations, namely sensing the targets, updating its search map, and sharing the information with neighbors.

Furthermore, each UAV is equipped with a wireless communication device that allows it to communicate with its neighbors, either broadcasting or receiving a message I through the two primitives $M_i(n, I)$ and $R_i(n, I)$ to spread and to receive the message I, respectively. Typically, such neighbors are assumed to be inside a circle of radius $\mathcal{R}$, centered in U_i, i.e., $\mathcal{N}_t(U_i) = \left\{U_j, \| U_i, U_j \| \leqslant \mathcal{R}\right\}$ where $\| .,. \|$ denotes the Euclidean distance, as can be noticed in Fig. 2. That is, the communication becomes reliable as long as the neighbors are inside such a circle.

At a given time $t = m\tau$, U_i senses $c_{x,y}$, through a fixed rectangular footprint sensor downward-pointing characterized by an imperfect detection accuracy and communication capabilities, leveraging the primitive $S_i(x, y)$. As a result, the obtained observation is binary: (i) $O_{x,y}^{(i,t)} = 1$ indicating that the cell $c_{x,y}$ is visited at the time t by U_i and something has been observed, or (ii) $O_{x,y}^{(i,t)} = 0$ stating that U_i did not observe anything while flying at such a time on this cell, i.e., no target detection.

Due to the imperfectness of the sensor, an observation of the detected target still does not guarantee the presence of the target. The performance of the sensor can be described with two probabilities, namely the sensor detection rate p_i and the false alarm rate q_i, whose are time-independent, and chosen to be close to 1 and 0, respectively [13] as

$$\begin{cases} \Pr\left(O_{x,y}^{(i,t)} = 1 | X_{x,y}^{(i,t)} = 1\right) = p_i \\ \Pr\left(O_{x,y}^{(i,t)} = 0 | X_{x,y}^{(i,t)} = 1\right) = 1 - p_i, \end{cases} \tag{2}$$

$$\begin{cases} \Pr\left(O_{x,y}^{(i,t)} = 1 | X_{x,y}^{(i,t)} = 0\right) = q_i \\ \Pr\left(O_{x,y}^{(i,t)} = 0 | X_{x,y}^{(i,t)} = 0\right) = 1 - q_i, \end{cases} \tag{3}$$

where $1 - p_i$ and $1 - q_i$ denote the missed detection and true miss probabilities, respectively.

III. UAV DYNAMIC MODEL

During the exploration process, each U_i stores two individual maps:

- The map of target existence probabilities,

$$P^{(i)}(t) = \left\{P^{(i,t)}_{x,y}, x = 1..L_x\ ,\ y = 1..L_y\right\}, \tag{4}$$

- The cells counter map storing the number of visits per cells and UAVs up to a given time t,

$$C^{(i)}(t) = \left\{C^{(i,t)}_{x,y}, x = 1..L_x\ ,\ y = 1..L_y\right\}, \tag{5}$$

where $C^{(i,t)}_{x,y}$ accounts for the cumulated visits number of a cell by all UAVs until t, which must be incremented by one after each visit by a UAV. Typically, this variable is updated taking into account the maximum number of visits performed by the neighbors alongside the one of the current UAV until the previous time as

$$C^{(i,t)}_{x,y} = 1 + \max_{k \in \mathcal{N}_t(U_i)} \left\{C^{(i,t-\tau)}_{x,y}, C^{(k,t-\tau)}_{x,y}\right\}. \tag{6}$$

Every time step, each UAV has different probabilities of target existence into a given cell. Such probabilities are updated after each exploration of the cells with the help of Bayesian rule [32] as

$$P^{(i,t)}_{x,y} = \begin{cases} \dfrac{p_i P^{(i,t-\tau)}_{x,y}}{p_i P^{(i,t-\tau)}_{x,y} + q_i\left(1 - P^{(i,t-\tau)}_{x,y}\right)}, & O^{(i,t)}_{x,y} = 1 \\[2ex] \dfrac{(1-p_i) P^{(i,t-\tau)}_{x,y}}{(1-p_i) P^{(i,t-\tau)}_{x,y} + (1-q_i)\left(1 - P^{(i,t-\tau)}_{x,y}\right)}, & O^{(i,t)}_{x,y} = 0. \end{cases} \tag{7}$$

Furthermore, we have the following

$$P^{(i,t)}_{x,y} - P^{(i,t-\tau)}_{x,y} = \begin{cases} \dfrac{(p_i - q_i)\left(1 - P^{(i,t-\tau)}_{x,y}\right) P^{(i,t-\tau)}_{x,y}}{p_i P^{(i,t-\tau)}_{x,y} + q_i\left(1 - P^{(i,t-\tau)}_{x,y}\right)}, & O^{(i,t)}_{x,y} = 1 \\[2ex] \dfrac{-(p_i - q_i)\left(1 - P^{(i,t-\tau)}_{x,y}\right) P^{(i,t-\tau)}_{x,y}}{(1-p_i) P^{(i,t-\tau)}_{x,y} + (1-q_i)\left(1 - P^{(i,t-\tau)}_{x,y}\right)}, & O^{(i,t)}_{x,y} = 0. \end{cases} \tag{8}$$

Subsequently, since $p_i > q_i$ and $P^{(i,t)}_{x,y} \in [0, 1]$, it can be noticed that $P^{(i,t)}_{x,y} \geqslant P^{(i,t-\tau)}_{x,y}$ if $O^{(i,t)}_{x,y} = 1$ and $P^{(i,t)}_{x,y} \leqslant P^{(i,t-\tau)}_{x,y}$ if not.

The above probabilities update process is reiterated until the kth target's existence confirmation ($k \leq N_T$) in a specific $c_{x,y}$ cell characterized by the condition $P^{(j,t_k)}_{x,y} > \theta$, where $j \in \{i, \mathcal{N}_t(U_i)\}$ and θ is a predefined threshold chosen close to 1, i.e.,

$$t_k = \arg\min_t \left\{P^{(j,t)}_{x,y} > \theta, j \in \{i, \mathcal{N}_t(U_i)\}\right\}. \tag{9}$$

Under such a constraint, a target is most likely found by U_i at the aforementioned place. Indeed, it can be easily ascertained that the two sub-equations in (7) lead after certain steps to a unique solution which is $P^{(i,*)}_{x,y} = 1$. In fact, by replacing $P^{(i,t)}_{x,y}$ at any given time by the value at the convergence $P^{(i,*)}_{x,y}$, one can obtain

$$\begin{cases} P^{(i,*)}_{x,y} = \dfrac{p_i P^{(i,*)}_{x,y}}{p_i P^{(i,*)}_{x,y} + q_i\left(1 - P^{(i,*)}_{x,y}\right)} \\[2ex] P^{(i,*)}_{x,y} = \dfrac{(1-p_i) P^{(i,*)}_{x,y}}{(1-p_i) P^{(i,*)}_{x,y} + (1-q_i)\left(1 - P^{(i,*)}_{x,y}\right)}. \end{cases} \tag{10}$$

As a result,

$$\begin{cases} q_i(1 - P^{(i,*)}_{x,y}) = p_i(1 - P^{(i,*)}_{x,y}) \\ (1 - q_i)(1 - P^{(i,*)}_{x,y}) = (1 - p_i)(1 - P^{(i,*)}_{x,y}). \end{cases} \tag{11}$$

As $p_i \neq q_i$, it follows that $P^{(i,*)}_{x,y} = 1$.

For the sake of notation, the variables t_k are sorted in the increasing order of time, as can be ascertained in Fig. 3, namely

$$t_{i_1} \leq t_{i_2} \leq .. \leq t_{i_{N_T}}, \tag{12}$$

where $i_k \in \{1..N_T\}$.

Furthermore, once all the N_T targets have been found, the maximum exploration instance time of our algorithm can be retrieved as follows

$$I_{\max} = \max_{k \leq N_T} t_k, \tag{13}$$

while the average time of validating all targets can be evaluated as

$$\bar{t}_{N_T} = \frac{\sum_{k=1}^{N_t} t_k}{N_T}. \tag{14}$$

Importantly, either in PSO or *Greedy* mode, once the ith UAV flies into the area covered by some of its neighbors, i.e., $\{U_k\}_{k \in \mathcal{N}_t(U_i)}$, its target detection probability, evaluated in (7), is updated exclusively once $O^{(i,t)}_{x,y} = 1$ by averaging all the probabilities of such neighbors along with its own one as

$$P^{(i,t)}_{x,y} = \begin{cases} P^{(i,t-\tau)}_{x,y}, & O^{(i,t)}_{x,y} = 0 \\[1ex] \dfrac{P^{(i,t-\tau)}_{x,y} + \sum_{k \in \mathcal{N}_t(U_i)} P^{(k,t-\tau)}_{x,y}}{|\mathcal{N}_t(U_i)| + 1}, & O^{(i,t)}_{x,y} = 1. \end{cases} \tag{15}$$

The average probability of validating the target T_k is given by

$$\overline{P}^{(k)} = \frac{\sum_{j \in \{i, \mathcal{N}_{t_k}(U_i))\}} P^{(j,t_k)}_{x,y}}{\left|\mathcal{N}_{t_k}(U_i))\right| + 1}, \tag{16}$$

while the average probability of validating all targets can be obtained as follows

$$\overline{P} = \frac{\sum_{k=1}^{N_T} \overline{P}^{(k)}}{N_T}. \tag{17}$$

IV. PROBLEM-SOLVING ALGORITHM BASED ON LOCAL-PSO

In this section, the updated position of the UAV is provided in both considered modes in this paper. In our context, the particle is represented by a UAV e.g., U_i, where its position and velocity can be, respectively, modeled as

$$\chi_i^{(t)} = \left(\chi_{i,x}^{(t)}, \chi_{i,y}^{(t)}\right), \tag{18}$$

and

$$V_i^{(t)} = \left(V_{i,x}^{(t)}, V_{i,y}^{(t)}\right). \tag{19}$$

A. UPDATED POSITION UNDER THE GREEDY MODE

In the *Greedy* mode, each U_i moves with the same velocity $V_i^{(t)}$ from one cell to another by using the cells counter map to head towards less visited cells, i.e., the next cell to visit is characterized by

$$\chi_i^{(t)} = \arg \min_{j=0..2} C_{x_j,y_j}^{(i,t)}. \tag{20}$$

B. UPDATED POSITION UNDER LOCAL-PSO

It is noted that in the classical PSO algorithm, a population of individuals is initially generated with random positions and velocities in the search space. Each individual is considered as a particle, which is a potential solution. Every individual moves and evolves cognitively with other particles to seek the global optimum solution. Precisely, those motions are driven by their best position as well as the one of the swarm, determined by their fitness values. The particles in the PSO approach obey rules that are comparable to those that control the movements of UAVs in our suggested model. In fact, the algorithm's particle may be translated to a swarm member UAV, and the heuristic collaborative search can then be performed by duplicating the algorithm's principles while taking physical constraints into consideration [32].

In our proposed Local-PSO algorithm, all the UAVs start initially moving in the *Greedy* mode, until a target $T_k = (T_{k,x}, T_{k,y})$ is detected in the cell $c_{x,y}$ by U_i. This latter broadcasts the target detection information, and based on its neighbors $\mathcal{N}_t(U_i)$, build a local sub-swarm if $|\mathcal{N}_t(U_i)| \geqslant 3$, and launch the LoPSO algorithm. The sub-swarm will dissolve and all of its UAVs will proceed into *Greedy* mode search if the discovered target is successfully validated or the validation period has expired.

Each U_j, among the neighbors of U_i, receiving such an information decides its next flying mode according to the following scenarios:

- if the flight mode of U_j is *Greedy*, its next mode is set to PSO. Afterwards, its velocity and position are evaluated using PSO algorithm.
- if U_j already belongs to another sub-swarm, such information is ignored and didn't move to the U_i's sub-swarm.
- if U_j is cooperatively building another sub-swarm, it exits this latter if the number of its neighbors is less than 3. Next, it integrates the U_i's sub-swarm.

Let $f(x)$ be the fitness function defined as

$$f\left(\chi_i^{(t)}\right) = P_{x,y}^{(i,t)}. \tag{21}$$

It is worthwhile the greater the target existence probability is, the greater the fitness is. In two-dimensional space (2D), the best position of U_i at a given time t, i.e., $\delta_i^{(t)}$ is corresponding to the maximum value of its respective fitness until the time t as

$$f\left(\delta_i^{(t)}\right) = \max\left\{f\left(\chi_i^{(t)}\right), f\left(\delta_i^{(t-\tau)}\right)\right\}. \tag{22}$$

Furthermore, $\delta_i^{(0)}$ is initialized at the position of U_i once the sub-swarm is built, while the global best position of the sub-swarm $\mathcal{N}_t(U_i)$ at a given time t can be obtained relying on the local best positions defined in (22) as

$$\Delta_i^{(t)} = \left\{\delta_k^{(t)}, f\left(\delta_k^{(t)}\right) \geqslant \max_{j \neq k, j \in \{i, \mathcal{N}_t(U_i)\}} f\left(\delta_j^{(t)}\right)\right\}. \tag{23}$$

The LoPSO, launched by U_i on a certain sub-swarm, will be ended once the objective function $f\left(\Delta_i^{(t)}\right)$ exceeds θ. Afterwards, all the UAVs of this sub-swarm will be released from the LoPSO mode and start to search, using *Greedy* mode, another missed target.

In each iteration cycle, the velocity and position of U_i are updated, respectively, as follows [16]

$$V_{i,\mu}^{(t)} = r_0\omega^{(t-\tau)}V_{i,\mu}^{(t-\tau)} + r_1c_1V_{i,\mu}^{(t-\tau,l)} + r_2c_2V_{i,\mu}^{(t-\tau,g)}, \tag{24}$$

where

$$V_{i,\mu}^{(t-\tau,l)} = \frac{\delta_{i,\mu}^{(t-\tau)} - \chi_{i,\mu}^{(t-\tau)}}{\tau}, \tag{25}$$

$$V_{i,\mu}^{(t-\tau,g)} = \frac{\Delta_{i,\mu}^{(t-\tau)} - \chi_{i,\mu}^{(t-\tau)}}{\tau}, \tag{26}$$

and

$$\chi_{i,\mu}^{(t)} = \chi_{i,\mu}^{(t-\tau)} + V_{i,\mu}^{(t-\tau)}\tau; \tag{27}$$

where $\omega^{(0)}$ is chosen in the range $[1, 2]$, μ equals either x or y, r_0, r_1 and r_2 denote three random values referring to the acceleration coefficients chosen in the range $]0, 1[$. Further, c_1 and c_2 are two positive numbers chosen appropriately, as will be discussed in the sequel.

Obviously, (25) and (26) represent the velocity of U_i at the time $t - \tau$ if the UAV moves to the local or the global position provided by its sub-swarm, respectively. Importantly, (24) updates the velocity of the current UAV by averaging three main terms, namely:

i. its own velocity at the previous time, scaled by an appropriate factor $\omega^{(t-\tau)}$. Such a parameter is of paramount importance for convergence purposes, allowing to create a trade-off between the global and local exploration by the sub-swarm. Importantly, as $\omega^{(t)} = r_0\omega^{(t-\tau)}$ is a geometric series, it can be ascertained that for greater values of $\frac{t}{\tau}$, such a coefficient is tending to zero as $r_0 < 1$. Subsequently, with the time, such a velocity is computed almost by averaging exclusively the two last terms in (24). On the other hand, the three velocities appearing in (24) are initialized at the same value corresponding to the velocity of U_i when the sub-swarm is constructed.

Initially, for a given sub-swarm, the parameters $\omega^{(0)}$ is chosen above as the local and global positions are not yet determined.

ii. the expected velocity of U_i if it moved to its best local position at the time $t - \tau$. Again, this term is scaled by an appropriate cognitive component $c_1 r_1$;

iii. its expected velocity if it was moving at the time $t - \tau$ to the best global position provided by its sub-swarm. In a similar manner, such a term is scaled by a social factor $c_2 r_2$;

By carrying extensive representative simulations, our obtained results show that the optimum values of c_1 and c_2 shall be chosen in the range of [1.1, 1.4], whereas $\omega^{(0)}$ shall be set to max $\{c_1, c_2\}$.

At each time step, U_i shares exclusively the target detection information with its neighbors forming the same sub-swarms.

The cooperative and competitive method adopted in this paper avoids exploring the same target by two different sub-swarms at the same time. That is, only one target is sought by a fixed sub-swarm at a given time.

The process of our proposed method is carried out in two main algorithms launched separately in both base station and UAVs. Specifically, **BsMain**, and the other on the UAVs described by **FoundShare** function, **Neighbors** function, and **LoPSO** function.

C. BASE STATION PROCESS

BSMain presents the main algorithm running at the BS allowing to lunch the cooperative search. After establishing the probability and certainty maps, and providing the input parameters, such as the total number of targets and their random positions and velocities, as well as the initial position and speed and the flight mode of each UAV, i.e., *Greedy* or PSO, all the UAVs launch and start searching as long as the search time is not yet finished and there are still missing targets. The search continues until all targets are found or the search time is up, i.e., close to $D_{\max}$. In this case, all the UAVs will be alerted to stop the search. The BS listens to each UAV i continuously using $R_i(n, I)$ in order to gather information about the targets detected. If the received information I is about the detection of a target, i.e, $n = 3$, it will be examined to determine whether it is a new target in order to update the list of targets detected $\mathcal{T}$ and share this information with all UAVs through $M_0(4, \mathcal{T})$, otherwise, the message will be ignored. When all targets have been located, the BS instructs all UAVs to cease searching.

D. UAVs PROCESS

- At each U_i, the coordination update and information merging are illustrated in **FoundShare** function. The specific process is that the U_i starts searching targets in *Greedy* mode, as long as the end of the search has not yet been signaled by the BS and the energy is sufficient. To verify the existence of one target, the U_i gets sensing data from its sensors on the current cell $\chi_i(t)$ by $S_i(\chi_i(t))$ and updates its $P^{(i)}(t)$ and $C^{(i)}(t)$, then combines and shares its information with other neighboring UAVs.

  When a request to integrate a sub-swarm is made when the flying mode is *Greedy*, the U_i will accept the request and switch to *PSO* by sending an acceptance message to the issuing UAV. In case that one target is found in the hovered cell while in *Greedy* mode, the U_i switches to *PSO* mode and builds a sub-swarm using the **Neighbors** function, then runs the **LoPSO** function for target validation.

  Next, the BS will be alerted of the target's position once the detected target is verified, that is, if the LoPSO function return result is not equal to $(-1, -1)$, consequently the built sub-swarm will be released, and all its members switch to *Greedy* mode. In contrast, if the flying mode is *PSO*, the U_i stays integrated into the sub-swarm and flight in the *PSO* mode according to (24) and (27) and updating the search information as long as there is no release request.
- The construction of the set of neighbors is ensured by **Neighbors** function executed by the UAV which detects a target. The process consists to send an exploration message to nearby UAVs depending on the communication range and integrates whose response is True in the sub-swarm.
- Finally, the **LoPSO** function is the one that carries out the Local PSO approach, which orchestrates the movements of the different UAVs in the same sub-swarm according to PSO and validates the found target's position. Each UAV in the built sub-swarm must update its movement settings as well as the search information, then move to the new position, as long as the search time is less than the maximum D_{PSO}. When the goal function achieves the probability threshold, the PSO mode will terminate, and the swarm will dissolve, confirming the target's position.

V. SIMULATION RESULTS

In order to properly evaluate the proposed method and analyze its operation in the most realistic way possible, a set of simulation tests was realized to verify the behavior of the proposed approach, by modeling the UAVs and targets in a platform by setting the values of their parameters as indicated in Table 2. In our simulation, all experiments are based on $D_{\max} = 1000$ simulation runs.

To validate the effectiveness of our method, we compared our results with those obtained through related works: PSO, ACO, Greedy, MTS, and RS.

In the simulation experiments, and in order to use the relevant algorithm to make the swarm UAVs perform a search target in a given area, we must make the following assumptions:

TABLE 2. Parameters settings.

Parameter	*Value*
N_T	10
Target's velocity	$[0.8, 4]$ m/s
N_u	30
UAV's maximum velocity	10 m/s
p_i	$[0.8, 0.9]$
q_i	$[0.2, 0.3]$
$\mathcal{R}$	4
θ	0.97
L_x	50
L_y	50
h	100 m
$(r_i)_{i\geq 0}$ random values	$]0, 1[$
$(c_i)_{i\geq 1}$	1.25
$\omega^{(0)}$	1.2
τ	1 s
$D_{\max}$	1000 s
D_{PSO}	100 s

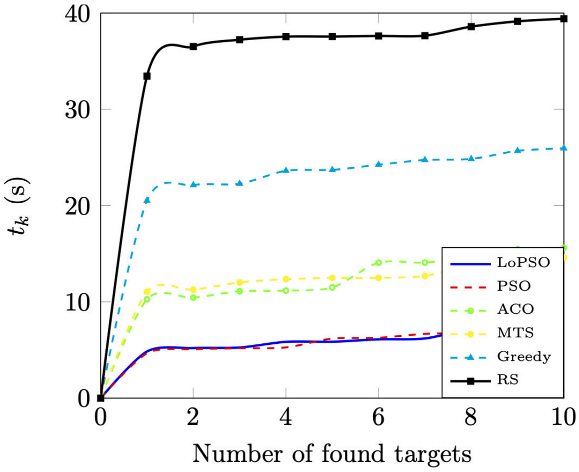


FIGURE 3. Duration of a found target' location confirmation.

(i) All targets can be detected. The main problem solved in this part is searching for targets in a certain area. Therefore, we assume that the target can be detected by the UAV as long as it appears within its detection range.

(ii) Location information can be shared between local swarm UAVs depending on the communication range.

(iii) The impact of bad weather and airflow on the stability of drones is not taken into account.

Tables 3 and 4 show the total search time of discovering all the targets, average time of validating target (14), and average probability (17) in the suggested approach when the number of targets and the size of the UAV swarm are increased, respectively. It can be clearly seen that almost all targets can be successfully located using the LoPSO method, with the exception of a few instances where the target cannot be found in extreme situations. In general, this method produces the best multi-target search results. Furthermore, in terms of search time, it is roughly proportional to the number of targets to be searched as noticed in Table 3. The success rate for all discovered targets has remained above 96 %.

Function Main Program

```
Function BSMain ($N_u$, $N_T$, $D_{\max}$, $\tau$):
    //Executed on the BS
    Initialization:
        $\mathcal{T} \leftarrow \phi$; //The set of found targets
        $N_d \leftarrow 0$; //The number of discovered targets
        $t \leftarrow 0$; //The current time
        $n \leftarrow 0$; //Indicates the type of received message
        $seek \leftarrow$ True; //Indicates that the search is in progress
        $M_0(0, \text{True})$; //Alert all the UAVs to continue searching
    for $i \leftarrow 1$ to $N_u$ do
        //launch all UAVs
        $FoundShare(i, t, N_u, D_{\max}, \theta)$;
    while $t < D_{\max}$ and $seek$=True do
        for $i \leftarrow 1$ to $N_u$ do
            $R_i(n, I)$; //sensing exist transmitted message $I$
            from all UAVs
            if $n$=3 then
                //the received message is a target position
                if $I \notin \mathcal{T}$ then
                    $\mathcal{T} \leftarrow \mathcal{T} \cup I$ ;
                    $N_d \leftarrow N_d + 1$ ;
                    $M_0(4, \mathcal{T})$;//share the set of discovered
                    target with all UAVs
        if $N_d = N_T$ then
            $seek \leftarrow$ False ;
            $M_0(0, \text{False})$;//Alert all the UAVs to stop the
            search
        $t \leftarrow t + \tau$ ;
    return $\mathcal{T}$;
```

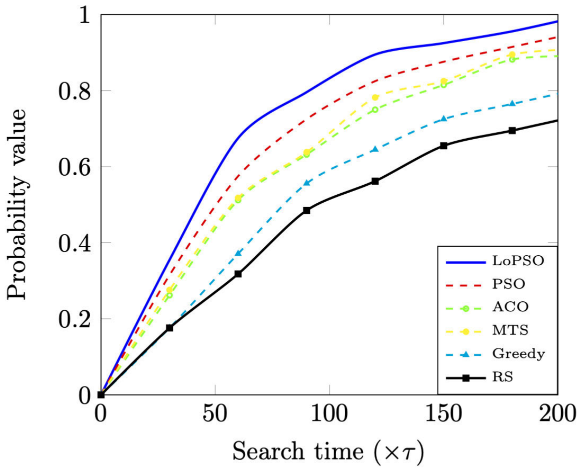


FIGURE 4. Existence Probability value of one found target' location vs search time.

Similarly, as indicated in Table 4, the suggested technique has a considerably better chance of detecting all of the targets. It can be observed that the suggested technique can locate all targets in any population with a swarm size larger than 10 with a probability that falls above 97 %. Indeed, with increasing UAV swarm size, the number of iterations until convergence reduces. Consequently, the method has a better

Function Found and Share

```
Function FoundShare($i$,$t$,$N_u$,$D_{\max}$,$\theta$):
    //Runs in $U_i$
    Initializing:
        $\mathcal{T} \leftarrow \phi$;//the $U_i'$s local set of discovered targets
        $D_{\text{PSO}}$,$V_i^{(t)}$, $\chi_i^{(t)}$, $E_t$, $E_i$, $p_i$, $q_i$, $P^{(i)}(t)$, $C^{(i)}(t)$;
        $t \leftarrow 0$;
        $n \leftarrow 0$;//Indicates the type of received message
        $F_i \leftarrow$ "Greedy";
        $Swarm \leftarrow$ False; //UAV′ response to either integrate
    or not a sub – swarm
        $L \leftarrow$ True; //indicates the search status progress
    while $L! =$ False and $E_i > E_t$ and $t < D_{\max}$ do
        $O_{x,y}^{(i,t)} \leftarrow S_i(\chi_i^{(t)})$;//Sense a target
        Update $P^{(i)}(t)$;//According to (7)
        $M_i(1, P^{(i)}(t))$;//Broadcast the probability map
        $M_i(2, C^{(i)}(t))$;//Broadcast the cells visited map
        $\mathcal{N}_t(U_i) \leftarrow Neighbors(i, t, N_u)$;
        if $F_i =$ "Greedy" then
            if $O_{x,y}^{(i,t)} = 1$ and $\chi_i^{(t)} \notin \mathcal{T}$ then
                $T \leftarrow LoPSO(i, \theta, t, D_{\text{PSO}}, \mathcal{N}_t(U_i))$;
                if $T \neq (-1, -1)$ then
                    //One target is confirmed
                    $M_i(3, T)$ ; //Share with ground station

            else
                for $j \leftarrow 1$ to $|\mathcal{N}_t(U_i)|$ do
                    $R_j(n, I)$;
                    if $n = 0$ and $I =$"Neighbors" then
                        //Request to join the sub – swarm
                        $Swarm \leftarrow$ True;//Accept the request
                        break;

                    else if $n = 1$ then
                        Update $P^{(i)}(t)$;//According to (15)

                    else if $n = 2$ then
                        Update $C^{(i)}(t)$;//According to (6)

                if $Swarm =$ True then
                    $M_i(0, \text{True})$;//Response by agreeing to
                    join the sub – swarm
                    $F_i \leftarrow$ "PSO";
                else
                    Update
                      $\chi_i^{(t)}$;//According to Greedy mode (20)
                    Move to $\chi_i^{(t)}$;

        else
            for $j \leftarrow 1$ to $|\mathcal{N}_t(U_i)|$ do
                $R_j(n, I)$;//Sensing exist transmitted message
                $I$ from neighbors
                if $n = 0$ and $I =$"Free" then
                    $F_i \leftarrow$ "Greedy";
                    break;

        $t \leftarrow t + \tau$;
        $E_i \leftarrow E_i - 1$;
        $R_0(n, I)$;//Sensing exist transmitted message from BS
        if $n = 4$ then
            //The message received is the set of discovered
            targets
            $\mathcal{T} \leftarrow I$;
        else if $n = 0$ then
            //The received message is about search status
            $L \leftarrow I$;
    return True; //UAV completes its mission
```

Function Neighbors

```
Function Neighbors($k$,$t$,$N_u$):
    //Constructing the neighbors set of the kth UAV
     Initialization
    $\mathcal{N}_t(U_k) \leftarrow \phi$; //Initialise the set indices of neighbors
    $M_k(0, \text{"}Neighbors?\text{"})$;
     //Invite all neighbors to join the sub – swarm
    for $j \leftarrow 1$ to $N_u$ do
        $R_j(n, I)$; //Sensing a potential transmitted message $I$
        from neighbors
        if $n = 0$ and $I =$ True then
            $\mathcal{N}_t(U_k) \leftarrow \mathcal{N}_t(U_k) \cup U_j$;
            //Approval received from $\text{U}_\text{j}$ to integrate the
            sub – swarm
    return $\mathcal{N}_t(U_k)$;
```

TABLE 3. Average search times, average validation time, and average validation probability (%) for different targets number.

N_T	*Total Search time (s)*	$\bar{t}_{N_T}$ (s)	$\bar{P}(\%)$
1	121.62	4.85	97.89
2	191.99	5.2	99.81
3	240.03	5.26	95.13
4	286.66	5.85	97.16
5	321.46	5.86	99.57
6	360.66	6.11	96.26
7	398.80	6.2	97.53
8	442.35	7.19	99.29
9	504.10	7.26	99.39
10	650.78	7.37	97.85

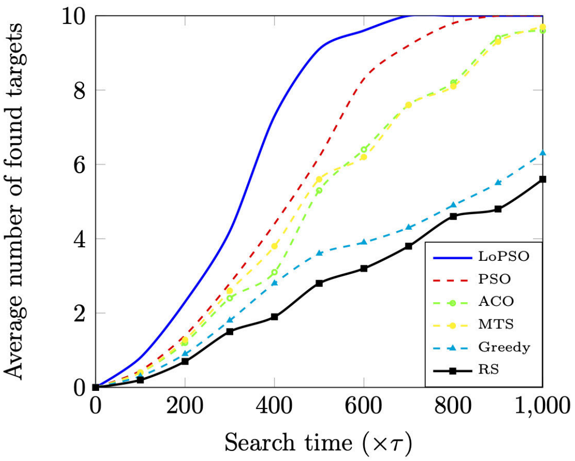


FIGURE 5. Number of detected targets vs search time.

likelihood of locating all the targets as the size of the UAV swarm grows.

As can be seen, the approach described in this study not only finds practically all targets but also has a shorter average iteration time. This method's efficacy is then proved, except for the situations with the smallest number of UAVs, e.g., $N_u = 10$ and the most extreme target dispersion, e.g., spread on the map corners, all targets may be identified in the other cases.

Function Local PSO

```
Function LoPSO($(i, \theta, t, D_{\text{PSO}}, \mathcal{N}_t(U_i))$):
    Initializing:
    $t_i \leftarrow t$ ;
    $F_i \leftarrow$ "PSO";
    $PSO \leftarrow$True ;
    while PSO=True and $t - t_i \leqslant D_{\text{PSO}}$ do
        if $|\mathcal{N}_t(U_i)| \geqslant 3$ then
            foreach $U_j \in \mathcal{N}_t(U_i)$ do
                Compute $\chi_j^{(t)}$; //According to (27)
                Compute $\delta_j^{(t)}$;//According to (22)
                Compute $V_j^{(t)}$; //According to (24)
                Update $P^{(j)}(t)$; //According to (7) and (15)
                Update $C^{(j)}(t)$; //According to (6);
                Compute $\Delta_i^{(t)}$;//According to (23)
                Move to $\chi_j^{(t)}$;
                if $f(\Delta_i^{(t)}) \geqslant \theta$ then
                    $F_i \leftarrow$ "Greedy";
                    foreach $U_j \in \mathcal{N}_t(U_i)$ do
                        $M_i(0,$ "Free");
                    $\mathcal{N}_t(U_i) \leftarrow \phi$;
                    return $\Delta_i^{(t)}$;
                else
                    Compute $\Delta_i^{(t)}$;//According to (23)
                    $t \leftarrow t + \tau$ ;
        else
            $F_i \leftarrow$ "Greedy";
            $\mathcal{N}_t(U_i) \leftarrow \phi$;//Initialize the neighbors set
            $PSO \leftarrow$False ;
    if $t - t_i > D_{\text{PSO}}$ then
        $F_i \leftarrow$ "Greedy";
        foreach $U_j \in \mathcal{N}_t(U_i)$ do
            $M_i(0,$ "Free");//Prevent each neighbor UAV to
            //breaking free from the sub − swarm
        $\mathcal{N}_t(U_i) \leftarrow \phi$;
        $PSO \leftarrow$False ;
        return $(-1, -1)$; // No target detected
```

TABLE 4. Average search times, average validation time, and average validation probability (%) under different UAVs number.

N_u	*Total Search time (s)*	$\bar{t}_{N_T}$ (s)	$\overline{P}(\%)$
5	6000	6000	15.25
10	1638.24	13.66	97.63
15	1593.76	14.08	99.72
20	986.51	5.18	99.83
25	753.84	6.27	99.91
30	650.78	5.87	100.00
35	480.09	5.73	100.00
40	492.45	5.68	100.00
45	447.84	5.57	100.00
50	416.18	5.17	100.00

The total computational complexity of the proposed approach is given in Table 5. It is worthy to mention that the method's overall calculation complexity, i.e., that of the main function *BSMain*, is calculated based on the complexity of other functions: *Neighbors*, *LocalPSO*, and *FoundShare*.

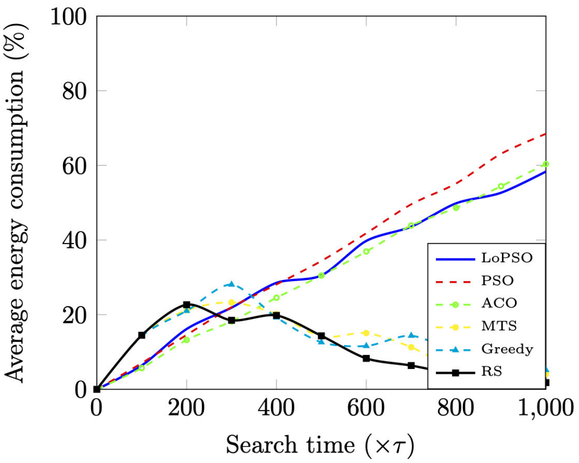


FIGURE 6. Communication energy consumption vs search time.

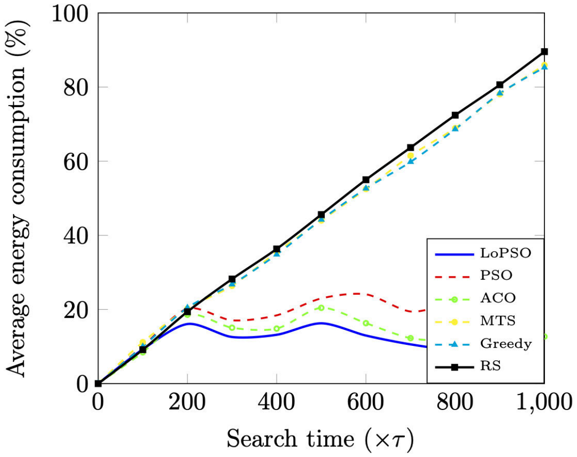


FIGURE 7. Mobility energy consumption vs search time.

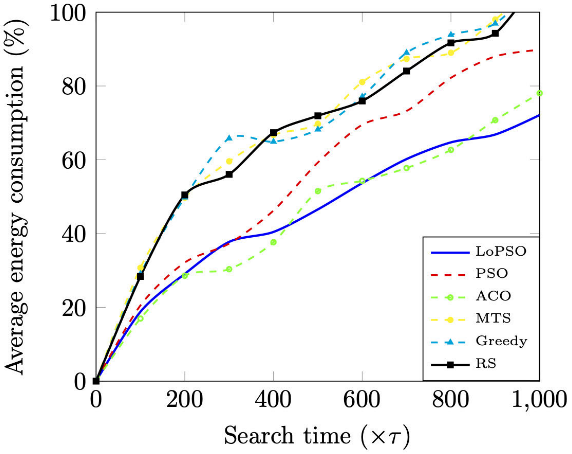


FIGURE 8. Energy consumption vs search time.

Fig. 3 shows the time needed to validate the detection of the kth ($1 \leq k \leq 10$) target using the six search methods with the help of (9). As mentioned in (12), such durations

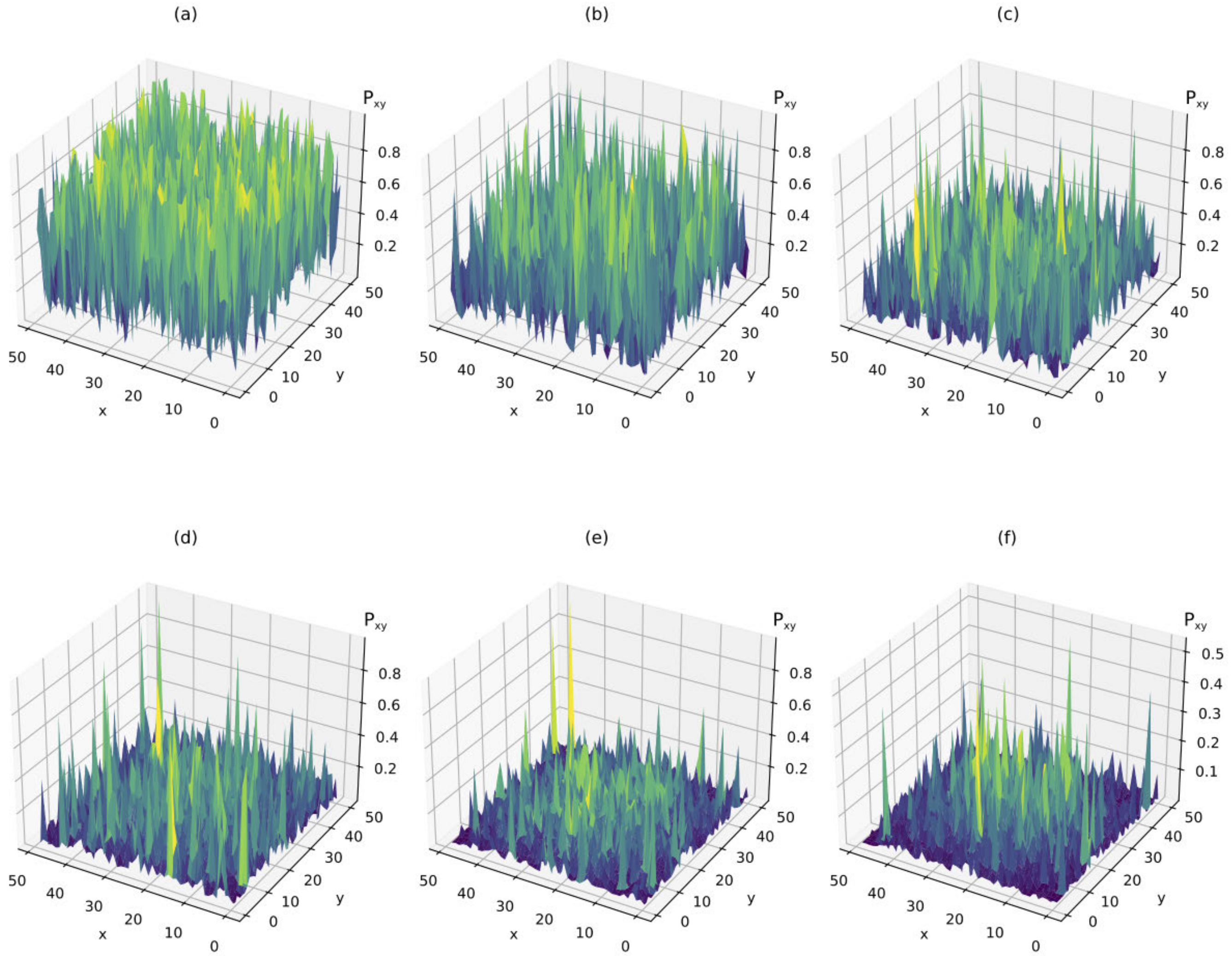


FIGURE 9. Convergence of a Probability map for a fixed UAV for $\mathcal{R} = 4$ cells. (a) $t = 0$ *s*. (b) $t = 200$ *s*. (c) $t = 400$ *s*. (d) $t = 600$ *s*. (e) $t = 800$ *s*. (f) $t = 1000$ *s*.

TABLE 5. Computational complexity of LoPSO algorithm.

Function	*Calculation time*	*Complexity*
Neighbors	$T_{Neighbors}(N_u) = 3N_u + 2$	$\mathcal{O}(N_u)$
LoPSO	$T_{LoPSO}(N_u) = 3 + \frac{D_{PSO}}{\tau}((N_u - 1)(7 + (N_u - 1) + 1))) + 1 + (N_u - 1) + 2$ $= \frac{D_{PSO}}{\tau}N_u^2 + (6\frac{D_{PSO}}{\tau} + 1)N_u + 4 - 7\frac{D_{PSO}}{\tau}$	$\mathcal{O}(N_u^2)$
FoundShare	$T_{FoundShare}(N_u, N_T) = 6 + N_T\left(10 + T_{Neighbors}(N_u) + T_{LoPSO}(N_u)\right))$ $= \frac{D_{PSO}}{\tau^2}N_T N_u^2 + 3(2\frac{D_{PSO}}{\tau} + 3)N_T N_u + 6(16 - 7\frac{D_{PSO}}{\tau})$	$\mathcal{O}(N_T N_u^2)$
Main	$T_{Main}(N_u, N_T) = N_T T_{FoundShare}(N_u, N_T)$	$\mathcal{O}(N_T^2 N_u^2)$

are sorted in an increasing order with respect to the number of found targets. It has been demonstrated that LoPSO and PSO provide for faster validation of target detection than the other techniques. We can see that the validation takes less than 5 runs steps, compared to other methods that take at least 15 runs steps. This is because the constructed local swarm for each target identified by one UAV allows the target to be swiftly validated by increasing the number of visits carried out by the local swarm's UAVs.

Fig. 4 illustrates the variation of the detection probability value according to the time for the six search methods. As depicted, for a $D_{\max} = 1000$ s, the target existence probability value rapidly reaches 0.925 for $t = 150$ s in the LoPSO process, which means in 12.25 % of the search time. Indeed, the proposed mechanism takes into account both the cooperation and the competition between the UAVs of the same sub-swarm when they are in the same cell, which makes it possible to increase the

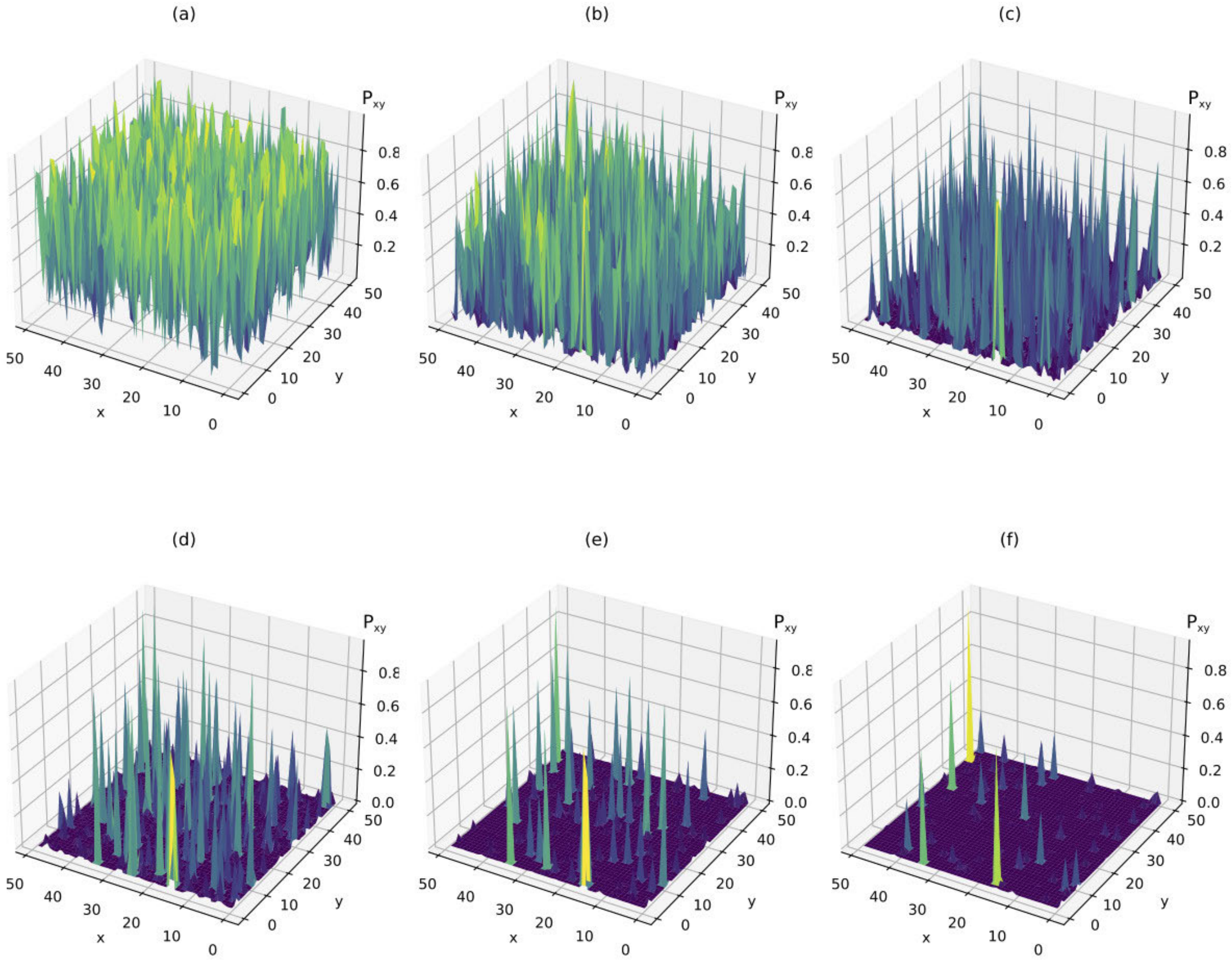


FIGURE 10. Convergence of a Probability map for a fixed UAV for $\mathcal{R} = 8$ cells. (a) $t = 0$ *s*. (b) $t = 200$ *s*. (c) $t = 400$ *s*. (d) $t = 600$ *s*. (e) $t = 800$ *s*. (f) $t = 1000$ *s*.

number of target detection by these UAVs in a reduced time search.

Fig. 5 illustrates, for several exploration algorithms, the number of found targets as a function of search time. It is obvious that the LoPSO method enhances the targets' search performance, as it can be noticed that roughly 50 % of the targets were detected in about 30 % of the search time, whereas all targets were detected in about 50 % of the full run time.

Fig. 8 depicts the average energy consumption usage as a function of search time. We take into account here, both consumption due to communication and mobility. These findings show that energy usage rises over time and it can be observed that the proposed LoPSO approach has a lower consumption than the other methods. Indeed, on the mobility side, because the UAVs move at the same speed towards the next cell in the three modes RS, Greedy, and MTS, the mobility energy consumption will increase linearly. However, with PSO, LoPSO, and ACO, the speed of the UAVs was determined by the swarm, which implies that it is not constant, and therefore the mobility energy consumption was not linear as can be noticed in Fig. 7. PSO and ACO, on the other hand, have a linear increase in communication energy consumption owing to the policy set for the construction of a single swarm, which needs the same number of UAVs inside it and, as a consequence, the same number of messages exchanged between the drones each time step. However, the UAVs only exchange information when it is required in the three aforementioned modes, implying that communication energy usage is not linear as can be seen in Fig. 6.

In Figs. 9-10, the improvement of target detection probabilities versus the search time is depicted for various values of total search duration for two values of the communication range of a single UAV. We can notice that the convergence speed of the probability map for rapid target detection is higher with a wider communication range (Fig.9, $\mathcal{R} = 4$ cells and Fig. 10, $\mathcal{R} = 8$ cells) because it allows summoning a large number of UAVs to build a large sub-swarm, which increases the search coverage. It can be seen that for $t = 1000$ s most of the targets are detected and located Fig. 10(f). However, in Fig. 9(f), there is still a lot of uncertainty about the targets' positions.

VI. CONCLUSION

In this paper, a new approach based on Local PSO for the optimal search of a set of moving targets with the help of multiple UAVs is presented. Such an algorithm allows dynamic management of the construction of local sub-swarms depending on the target detection status. The limitations of UAVs and the constraints of the search algorithm were considered, such as the imperfect sensing. The LoPSO has three advantages, which result in a higher level of performance compared to other concurrent approaches. Firstly, depending on the detection state of a target, the local sub-swarm built makes it possible to increase the number of hovers above the target, which helps to reduce the needed search time to validate its position. Secondly, the fact of releasing the drones after the validation of a target makes it possible not to monopolize all UAVs for a single target as in PSO, and consequently to go looking for others. Finally, the mechanism for creating local sub-swarms made it possible to reduce the communication flow since communication is limited, and only possible between UAVs of the same sub-swarm. To verify the performance of the proposed algorithm, several experiments were carried out in a simulation. The experimental results show that the performance of LoPSO is better than that of other approaches especially in terms of the number of targets validated in the minimum search time. As perspectives for future work, we aim at studying the LoPSO's benchmarking functions and exploring its ability to solve other complex optimization problems, e.g., the UAVs' cooperative search path planning with heterogeneous sensors.

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, Jan. 2021.

[3] W. Yang, J. Tang, R. He, and Y. Chen, "A medium-term conflict detection and resolution method for open low-altitude city airspace based on temporally and spatially integrated strategies," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 5, pp. 1817–1830, Sep. 2020.

[4] S.-P. Hong, S.-J. Cho, M.-J. Park, and M.-G. Lee, "Optimal search-relocation trade-off in Markovian-target searching," *Comput. Oper. Res.*, vol. 36, no. 6, pp. 2097–2104, Jun. 2009.

[5] M. Alighanbari and J. P. How, "Cooperative task assignment of unmanned aerial vehicles in adversarial environments," in *Proc. Amer. control Conf.*, vol. 7, Portland, OR, USA, Jun. 2005, pp. 4661–4666.

[6] W. Zhao, Q. Meng, and P. W. H. Chung, "A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario," *IEEE Trans. Cybern.*, vol. 46, no. 4, pp. 902–915, Apr. 2016.

[7] X. Guangtong, L. Li, T. Long, Z. Wang, and M. Cai, "Cooperative multiple task assignment considering precedence constraints using multi-chromosome encoded genetic algorithm," in *Proc. AIAA Guid., Navigat., Control Conf.*, Kissimmee, FL, USA, Jan. 2018, pp. 1–9.

[8] N. Buckman, J. P. How, and H. L. Cho, "Partial replanning for decentralized dynamic task allocation," in *Proc. AIAA Guid., Navigat. Control Conf.*, San Diego, CA, USA, Jan. 2019, p. 915.

[9] Z. Yao, M. Li, Z. Chen, and R. Zhou, "Mission decision-making method of multi-aircraft cooperatively attacking multi-target based on game theoretic framework," *Chin. J. Aeronaut.*, vol. 29, no. 6, pp. 1685–1694, Dec. 2016.

[10] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian, "Deep decentralized multi-task multi-agent reinforcement learning under partial observability," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Sydney, NSW, Australia, Jun. 2017, pp. 2681–2690.

[11] I. Kim and J. R. Morrison, "Learning based framework for joint task allocation and system design in stochastic multi-UAV systems," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Dallas, TX, USA, Jun. 2018, pp. 324–334.

[12] H. Saadaoui and F. E. Bouanani, "Information sharing in UAVs cooperative search based on calculating the minimum time," in *Proc. Int. Conf. Smart Digit. Environ.*, Rabat, Morocco, Jul. 2017, pp. 168–173.

[13] A. G. Shem, T. A. Mazzuchi, and S. Sarkani, "Addressing uncertainty in UAV navigation decision-making," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 1, pp. 295–313, Jan. 2008.

[14] M. L. Cummings, "Operator interaction with centralized versus decentralized UAV architectures," *Handbook Unmanned Aerial Vehicles*. Berlin, Germany: Springer, 2015, pp. 977–992.

[15] A. Karimoddini, M. Karimadini, and H. Lin, "Decentralized hybrid formation control of unmanned aerial vehicles," in *Proc. Amer. Control Conf.*, Portland, OR, USA, Jun. 2014, pp. 3887–3892.

[16] J. Liu, T. Shi, P. Li, X. Ren, and H. Ma, "Trajectories planning for multiple UAVs by the cooperative and competitive PSO algorithm," in *Proc. IEEE Intell. Veh. Symp.*, Seoul, South Korea, Jun. 2015, pp. 107–114.

[17] 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," *Renew. Energy*, vol. 113, pp. 266–280, Dec. 2017.

[18] H. Saadaoui and F. El Bouanani, "Information sharing based on local PSO for UAVs cooperative search of unmoved targets," in *Proc. Int. Conf. Adv. Commun. Technol. Netw. (CommNet)*, Marrakech, Morocco, Apr. 2018, pp. 1–6.

[19] Y. Wu, K. H. Low, and C. Lv, "Cooperative path planning for heterogeneous unmanned vehicles in a Search-and-Track mission aiming at an underwater target," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6782–6787, Jun. 2020.

[20] K. Hou, Y. Yang, X. Yang, and J. Lai, "Distributed cooperative search algorithm with task assignment and receding horizon predictive control for multiple unmanned aerial vehicles," *IEEE Access*, vol. 9, pp. 6122–6136, 2021.

[21] P. B. Sujit and D. Ghose, "Search using multiple UAVs with flight time constraints," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 2, pp. 491–509, Apr. 2004.

[22] M. Mirzaei, F. Sharifi, B. W. Gordon, C. A. Rabbath, and Y. M. Zhang, "Cooperative multi-vehicle search and coverage problem in uncertain environments," in *Proc. IEEE Conf. Decis. Control Eur. Control Conf.*, Orlando, FL, USA, Dec. 2011, pp. 4140–4145.

[23] S. Perez-Carabaza, E. Besada-Portas, J. A. Lopez-Orozco, and J. M. de la Cruz, "Ant colony optimization for multi-UAV minimum time search in uncertain domains," *Appl. Soft Comput.*, vol. 62, pp. 789–806, Jan. 2018.

[24] J. M. Hereford, M. Siebold, and S. Nichols, "Using the particle swarm optimization algorithm for robotic search applications," in *Proc. IEEE Swarm Intell. Symp. (SIS)*, Honolulu, HI, USA, Apr. 2007, pp. 53–59.

[25] J. Tang, G. Liu, and Q. Pan, "A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends," *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 10, pp. 1627–1643, Oct. 2021.

[26] Y. Du, "A novel approach for swarm robotic target searches based on the DPSO algorithm," *IEEE Access*, vol. 8, pp. 226484–226505, 2020.

[27] J. Sánchez-García, D. G. Reina, and S. L. Toral, "A distributed PSO-based exploration algorithm for a UAV network assisting a disaster scenario," *Future Gener. Comput. Syst.*, vol. 90, pp. 129–148, Jan. 2019.

[28] M. D. Phung and Q. P. Ha, ''Motion-encoded particle swarm optimization for moving target search using UAVs,'' *Appl. Soft Comput.*, vol. 97, no. 2, pp. 106705–106716, Sep. 2020.

[29] J. Yang, X. Wang, and P. Bauer, ''Extended PSO based collaborative searching for robotic swarms with practical constraints,'' *IEEE Access*, vol. 7, pp. 76328–76341, 2019.

[30] L. F. Bertuccelli and J. P. How, ''Search for dynamic targets with uncertain probability maps,'' in *Proc. Ameri. Cont. Conf.*, Minneapolis, MN, USA, Jun. 2006, pp. 737–742.

[31] L. F. Bertuccelli and J. P. How, ''Robust UAV search for environments with imprecise probability maps,'' in *Proc. 44th IEEE CDC-ECC*, Seville, Spain, Dec. 2005, pp. 5680–5685.

[32] J. Hu, L. Xie, K. Y. Lum, and J. Xu, ''Multiagent information fusion and cooperative control in target search,'' *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1223–1235, Jul. 2013.

[33] Z. Ren, T. Chen, and Z. Wu, ''Optimal matching control of a low energy charged particle beam in particle accelerators,'' *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 2, pp. 460–470, Mar. 2019.

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 is currently pursuing the Ph.D. degree with the ENSIAS College of Engineering, Mohammed V University of Rabat, Rabat. He has authored several publications in well-known conferences, such as COMMNET, ICSDE, SITA, SADEXS, CMT, and CIFEM. Since 2002, he has been with the preparatory classes for the high engineering schools ER-RAZI, El jadida, where he is currently a Computer Science Teacher. His current research interests include the 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, IT/ENSIAS College of Engineering, Mohammed V University, Rabat, in 2009, where he is currently an Associate Professor. He advised many Ph.D. and master's students at Mohammed V University and the University of Moulay Ismail. So far, his research efforts have culminated in more than 90 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 and the General Co-Chair for ACOSIS 2016 and CommNet 2018 conferences. He served as the TPC Chair for the ICSDE conferences, the General Co-Chair for ACOSIS'16, and the General Chair for the 2019/2020/2021 CommNet conferences. He is an Associate Editor of IEEE Access and *Frontiers in Communications and Networks* journal. He is also serving as a Lead Guest Editor for the Physical Layer Security Special Issue—*Frontiers in Communications and Networks* journal.

ELMEHDI ILLI (Member, IEEE) was born in Rabat, Morocco, in 1992. He received the B.Sc. and M.Sc. degrees in computer sciences, telecommunications, and electrical engineering from the Faculty of Sciences, Mohammed V University of Rabat, Rabat, in 2013 and 2015, respectively, and the Ph.D. degree from the ENSIAS College of Engineering, Mohammed V University of Rabat, in 2019. From 2018 to 2020, he served as a Teaching Assistant at the ENSIAS for undergraduate engineering students. Since January 2021, he has been a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, Khalifa University, Abu Dhabi, United Arab Emirates. He has authored several publications in well-known conferences and journals, such as WINCOM, IWCMC, GLOBECOM, IEEE Access, IEEE Transactions on Sustainable Computing, IEEE Open Journal of the Communications Society, and IEEE Transactions on Vehicular Technology. His main research interests include the performance analysis and the design of wireless communication systems, particularly optical-based communications, underwater optical communications, power-limited communications, signal processing, information theory, and physical layer security. He is a Reviewer in various prestigious journals, such as IEEE Access, *IEEE Communications Magazine*, IEEE Transactions on Communications, and IEEE Transactions on Green Communications and Networking. He served as an Organizing Committee Member for various international conferences, such as ACOSIS'16, CommNet'18, and CommNet'19. He was a recipient of the Best Paper Award from the ACOSIS'16 Conference. He served as the Registration, Website, and Media Chair for CommNet'20 Conference.