Robotica: (2021) volume 39, pp. 1909–1925. © The Author(s), 2021. Published by Cambridge University Press.
doi:10.1017/S0263574721000059

Search Strategies and Specifications in a Swarm versus Swarm Context

Ali Moltajaei Farid†‡*, Md Abdus Samad Kamal¶ and Simon Egerton§

†*School of Information Technology, Monash University, Subang Jaya, Malaysia*
‡*Monash Swarm Robotics Laboratory, Monash University, Clayton Campus, Melbourne, VIC 3800, Australia*
¶*Division of Mechanical Science and Technology, Graduate School of Science and Technology, Gunma University, Kiryu 376-8515, Japan*
E-mail: MASKAMAL@ieee.org
§*Department of Computer Science, Electrical Engineering, La Trobe University, Bendigo, Australia*
E-mail: S.EGERON@latrobe.edu.au

(Accepted January 8, 2021. First published online: March 2, 2021)

SUMMARY
This paper proposes and evaluates swarming mechanisms of patrolling unmanned aerial vehicles (UAVs) that can collectively search a region for intruding UAVs. The main contributions include the development of multi-objective searching strategies and investigation of the required sensor configurations for the patrolling UAVs. Numerical results reveal that it is sometimes better to search through a region with a single swarm rather than multiple swarms deployed over sub-regions. Moreover, a large communication range does not necessarily improve search performances, and the patrolling swarm must have a speed close to the speed of the intruding UAVs to maximize the search performances.

KEYWORDS: Swarm; Search; UAVs; Swarm versus swarm; Optimization.

1. Introduction

Unmanned aerial vehicles (UAVs), also popularly known as drones, are opening new applications in the commercial sectors, such as parcel delivery, emergency response, surveillance, aerial photography-video services, and many others. Besides these commercial applications, groups and individuals use drones privately, thanks to the ubiquity of low cost and quickly evolving technologies.[1] These civilian drones must follow some strict rules–regulations, for example, staying away from restricted zones or following predefined paths.[2,3] Furthermore, since the increased use of drones continues in the civilian sectors, various concerns about guarding individual rights and privacy invasions are emerging.[4,5] As the cases of violating the relevant laws and regulations are sometimes found in the news,[6] it is obvious that such violations might be widespread when the number of drones and their applications increase further. Therefore, some sort of monitoring or patrolling mechanism is required to keep the sky safe and secured. This paper addresses such an emerging need and proposes a new search framework using a set of drones that can collectively disperse in the target space to locate and track individuals or groups (swarms) of civilian drones. In other words, we develop a collective search mechanism using a set of drones, where the targets are also dynamic objects in 3D space.

1.1. Related work

Drones are considered to be very effective remote sensing tools, particularly in areas that traditionally have been inaccessible. Application areas include surveillance,[7] search and rescue,[8] natural disaster

* Corresponding author. E-mail: ali.farid@monash.edu

operations,[9] mapping[10] water sampling,[11] and inspection.[12] Recent research on remote sensing is mainly focused on the detection and tracking of objects located on the ground. Such a task is comparatively less complicated than detecting and tracking flying objects due to the vast field of view (FOV) of drones and the limited speed of ground moving objects. The remaining parts of this section briefly review current state-of-the-art technologies that can be used to enable drones to carry out search tasks.

Planning is a primary task for effective search. Various approaches to planning are proposed in the literature, for example, graph theory can be used to determine controlled random paths for effective search,[13] while stochastic models can be used for planning the paths of both single and multiple UAVs.[14] Hierarchical probabilistic decisions-based search for ground objects has also been proposed for multiple UAVs and unmanned ground vehicles.[14]

A set of paths is generated for the drone that ensures a complete sweep of the target surveillance area, which cleans the dirty space from intruder drones while simultaneously preventing new drones to enter.[15] This is a guaranteed but costly approach that requires a large number of drones for a large area, and therefore may not be feasible in the most practical scenarios.

Some recent works applied bio-inspired heuristic techniques for realizing intelligent search of the ground objects, for example, pedestrians or cars.[16–19] The firefly algorithm along with decentralized clustering was proposed for pedestrians tracking in a large crowd using UAVs.[16] Imitating the technique of how the living species explore an area for foods, an odor-centered search method was proposed by employing multiple mobile robots.[17] In a paper,[20] the robots search for wind direction and then search for the tails of odor to locate the plume source. In ref. [21], Hoff et al. used virtual pheromone in a 2D environment to enable the robots to choose a beacon or wandering tasks for foraging.

Over the years, pursuit-evasion games have been studied under various search aspects. With global visibility for all players in the game framework, they are not well-suited for robots with limited sensing capabilities in realistic scenarios. Perhaps, the probabilistic pursuit-evasion approach and dynamic programming[22] are the most widely used search methods in the robotic area. However, both methods are computationally demanding, and they could not fit well in the environment having large numbers of entities.[23] In a multi-robot system, decentralized control platforms are used to accomplish the search tasks.[24] However, in the literature, except a few,[25,26] searchers and evaders are only designed to undertake pursuit-evasion context in the 2D space, and their implementability and performance in a 3D environment remain unexplored.

For searching the 3D airspace, traditionally, ground station-based technologies, for example, radar or radar hybrid systems, have been used..[27–30] With emerging consumers for drones in the civilian sectors, these ground station-based technologies cannot be considered as an effective option because they are typically restricted to geographical sites. To overcome the above-mentioned issues, new flexible and mobile technologies are developed such as Skynet,[1] Droneshield,[2] and SkySafe.[3] However, they also have limitations when the number of drones is large.

Despite constraints in geographical sites, a swarm of UAVs is highly adaptable to different locations, providing low-cost portable means of monitoring. In the case of a large number of drones, for example, a swarm of drones, another set of searching UAVs can be employed to find and track the others, similar to the game theory. Such existing studies mostly rely on unlimited ranges of a camera and wireless communication,[31] except for some recent research.[32]

In ref. [30], authors used a swarm of submarines for exploring unmanned submarine where the main contribution is designing a swarm search that can work in ocean waves. Recently, Davis et al.[33] used a distributed Reynolds flocking model[34] based on a single virtual leader that influences the other members. The system in ref. [33] applied this model to fixed-wing UAVs for motion planning. In our earlier work,[35] a multi-objective heuristic method was proposed for proper search planning in the 3D space.

The existing search approaches, to the best of authors' knowledge, are sensitive to confusing behaviors of the drones.[28] The confusing phenomenon is a result of limited FOV of the searchers

[1] http://anti-drones.net/.

[2] https://www.droneshield.com/.

[3] https://www.skysafe.io/.

https://doi.org/10.1017/S0263574721000059 Published online by Cambridge University Press

and due to the intelligent behavior of the opponents who intentionally perplex the searcher to avoid being caught up. To the best of authors' knowledge, there is no reported intrusion by a swarm of flying intruders, and relevant research is very limited. We address a perceived future problem in the area of flying intruders. The main challenge in the search task is to detect the intruders with intelligent behavior against the searchers. Confusing behavior is one of such a bit of intelligence that is unknown to the searchers. The intruders may enhance their perception range beyond the physical FOV by sharing information via wireless communication and try to stay away from the searchers. The searchers need to be cooperative and intelligent with better physical capabilities (sensing range, speed, etc.) to tackle the intruders effectively.

Most of the existing search methods do not consider the impact of various factors related to practical implementation, for example, sensor configuration, limitation of FOV, communication range, or sudden failure of some UAVs. There are a few exceptions, in ref. [36] tracking using quadrotors that is limited to a laboratory experiment, while in refs. [32, 37] authors performed outdoor experiments with a swarm of 50 fixed-wing UAVs, but their system still requires a human in the loop for selecting the appropriate tactics.

1.2. Contribution

In this paper, we have proposed and developed searching methods to detect the moving drones in the 3D space by employing a swarm of monitoring drones.[38] Specifically, more structured search strategies are formulated that enable the monitoring swarm effectively utilizes their limited resources in a large area. These strategies are based on multi-objective optimization (MOO) that is implemented in a distributed but cooperative framework, where the individual monitoring drones share their states with their neighbors. The proposed strategies have been tested in a very realistic scenario where the drones have confusing-evasive behavior. Despite such complexity with limited resources, it is found that the proposed strategies could succeed in detecting the most targets within the reasonably minimum time with high efficiency. Furthermore, the proposed strategies have also been compared in terms of their continuing task accomplishment and final outcomes. Finally, the effects of the sensor configuration and the number of swarm members have also been evaluated and illustrated.

To address the issues mentioned above, in Section 2 system design is described, while Section 3 provides implementation-specific parameters and results from simulations that evaluate the performance of the presented algorithm. Section 4 concludes the paper and proposes future directions of research.

2. System Design

Consider a bounded 3D environment that contains an unknown number of UAVs or is subject to an incursion of unwanted or prohibited UAVs. These anonymous drones may have very simple or deterministic behavior or may have cooperative, intelligent, or evasive behavior. Consider that the environment contains some UAVs having heterogeneous behavior that form a swarm $\mathcal{H}$. Each UAV in swarm $\mathcal{H}$ is equipped with a camera having 360° FOV and a wireless communication device having limited signal transmission range R. In this sense, swarm $\mathcal{H}$ has a partial view of its environment.

For the purpose of monitoring such a swarm of unknown size, capability, and unknown heterogeneous behavior, we propose to employ another swarm of UAVs, named as monitoring or patrolling swarm $\mathcal{M}$, that can effectively disperse, search, and track swarm $\mathcal{H}$ to accomplish any given task. Considering practical aspects and cost-effectiveness, UAVs of swarm $\mathcal{M}$ are equipped with a simple camera mounted on a pan-tilt gimbal that has a limited FOV and a limited range R of wireless communication. Despite considering such limitations, an adaptive intelligent swarming mechanism is developed to accomplish the monitoring task efficiently. Specifically, the collective objective of the monitoring swarm, in a broad view, is to detect all the members of $\mathcal{H}$ efficiently by searching the space in a reasonably short time. The overall mission of the monitoring swarm should consist of dispersion, search, and tracking. However, the scope of this paper is kept limited to searching the space and detecting the unknown members of swarm $\mathcal{H}$ only. We have made a few assumptions in this fundamental research. Each UAV in the patrolling swarm has a limited communication range. The weather condition is good enough to detect any intruders when they come within a limited FOV. Furthermore, there is no hill or high-rise building that can limit the view or restrict the movement of the UAVs. Environmental conditions may affect the UAVs' search performance, but since this is fundamental research, we only consider ideal conditions.

https://doi.org/10.1017/S0263574721000059 Published online by Cambridge University Press

2.1. *Problem formulation*

Let us define the swarms using their individual members as $\mathcal{H} = \{h_1, h_2, \ldots, h_J\}$ and $\mathcal{M} = \{m_1, m_2, \ldots, m_I\}$. Consider a discrete-time framework, where each step is denoted by t and the step size is Δt. The position $p_q(t)$ of any UAV $q \in \{\mathcal{H}, \mathcal{M}\}$ (i.e., belongs to either swarm) at time t is expressed by the 3D coordinate with respect to the environment considered, which is given by $p_q(t) = [x_q(t), y_q(t), z_q(t)] \in \mathbb{R}^3$. Similarly camera orientation $f_q(t)$ of any UAV in Euler angels is given by $f_q(t) = [\theta_q(t), \phi_q(t), \psi_q(t)] \in \mathbb{R}^3$, where θ_q, ϕ_q, and ψ_q express yaw, pitch, and roll of the camera, respectively. The collective state of an UAV is given by $s_q = [p_q, f_q]$. The velocity of each UAV is given by

$$v_q(t) = \sqrt{v_{xq}^2(t) + v_{yq}^2(t) + v_{zq}^2(t)},$$

where v_{xq}, v_{yq}, and v_{zq} are vector elements of v_q. Furthermore, $v_q(t)$ is subject to a constraint $V_q^{\min} < v_q(t) < V_q^{\max}$.

At each time t, UAV $q \in \{\mathcal{M}, \mathcal{H}\}$ at state $p_q(t)$ can cover a volume V_q. For $q \in \mathcal{M}$, the covered volume is $V_m(f_q) = \alpha\beta\gamma/3$, and similarly, for $q \in \mathcal{H}$ centering at p_q, the covered volume is $V_h(f_q) = 4/3\pi r^3$, where α, β, γ, and r are diagonal, horizontal, vertical FOVs, and a camera range, respectively. The total covered volume by swarm $\mathcal{M}$ is $V_m = \cup_{m_i \in \mathcal{M}} f_i$, which is much smaller than the total volume of search space V_S ($V_m \ll V_S$).

Movement of UAV $q \in \{\mathcal{M}, \mathcal{H}\}$ from the current state to the next state is subject to the following constraints:

$$\|p_q(t+1) - p_q(t)\| < R_v, \;\; q \in \{\mathcal{H}, \mathcal{M}\}, \tag{1}$$

$$\|\theta_q(t+1) - \theta_q(t)\| \leq \frac{\pi}{24}, \;\; q \in \mathcal{M}, \tag{2}$$

$$\|\phi_q(t+1) - \phi_q(t)\| \leq \frac{\pi}{24}, \;\; q \in \mathcal{M}, \tag{3}$$

where R_v is a constant.

After planning, each UAV generates a proper control command to execute the planned motion according to its specifics physics. In the next section, we will describe how the swarm $\mathcal{H}$ members plan their motion for generating their behaviors.

2.2. *Multi-objective optimization*

2.2.1. Problem formulation. MOO is a tool that can handle different objectives simultaneously to choose the best action in a conflicting context. A brief review of MOO problem formulation and decision-making process are described here. First, let $S \in \mathbb{R}^3$ be a design space and $x = \{x_1(t), x_2(t), \ldots, x_n(t)\} \in S$ be the decision vector with lower and upper bounds given by $x_{\min} \leq x_i \leq x_{\max}$. The general MOO problem can be expressed using u objectives as

$$\min f(x) = \{f_1(x), f_2(x), \ldots, f_u(x)\}, \tag{4}$$

where $f_i(x): \mathbb{R}^n \to \mathbb{R}$ is the i-th objective. When there are conflicting objectives, improving one objective may deteriorate the others. To have the optimal value and identify a set of trade-off solutions, a Pareto front is used. A Pareto front is a set of all non-dominated solutions obtained from MOO. For the sake of defining the domination concept, candidate solutions are defined as $\{a, b\} \in S$. Candidate a dominates candidate b, that is, shown by $a \prec b$, if $\forall j = 1, \ldots, m \;\; f_i(a) < f_i(b) \wedge \exists\, j : f_j(a) \leq f_j(b)$, MOO identifies the closest approximation of true Pareto front, while MOO seeks for a diverse Pareto optimal set. A decision vector a^* is non-dominated or Pareto optimal if there is no other feasible decision vector $a \neq a^* \in S$ such that $f(a) \prec f(a^*)$.

2.2.2. Decision-maker. Some solutions on a Pareto front may work better than others due to the nature of the specific problem. The best solution must, therefore, be selected. This selection process can be performed in multiple ways. The most commonly used approaches are the knee-based approach and the vector-based approach. Knee-based approaches require at least four solutions on the Pareto front, and as shown in ref. [35], our system does not always satisfy this requirement. We

https://doi.org/10.1017/S0263574721000059 Published online by Cambridge University Press

have therefore opted for a vector-based approach.[39] We refer the reader to the appendix for a more detailed description of our vector-based approach.

2.3. *Behavior of swarm* ($\mathcal{H}$)

In practice, the behavior of swarm $\mathcal{H}$ is unknown and unpredictable; it may be a set of simple or very complex behaviors. Each planned movement of $h_j \in \mathcal{H}$ for time $t+1$ is represented by $p_j(t+1)$, and $p_j(t+1)$ represents the next position of UAV h_j, and we assume that h_j has $l = 1, .., n$ neighbors within its communication range. For the sake of considering the worst-case scenario, we assume swarm $\mathcal{H}$ has an intelligent complex behavior. We assume that swarm $\mathcal{H}$ follows subsequent objectives.

First of all, h_j UAVs consider an objective to keep large gaps from the other UAVs in $\mathcal{H}$. This objective is given by the following cost function, which is to be minimized:

$$c_1 = \sum_{j=1}^{n} \frac{1}{\|p_j(t+1) - p_l(t)\|}. \tag{5}$$

Swarm $\mathcal{H}$ produces complex motion to confuse swarm $\mathcal{M}$ and adds non-linear motion to their evasive trajectories using (6), which produces a zig-zag like motion. We define this objective as:

$$\begin{aligned} c_2 = {} & \textstyle\sum_{f=1}^{n} \frac{p_f(t) - p_j(t+1)}{(\|p_f(t) - p_j(t+1)\|)^2} \\ & + \sigma \textstyle\sum_{l=1}^{n} \frac{p_j(t+1) - p_l(t)}{(\|p_f(t) - p_j(t+1)\|)^2}, \end{aligned} \tag{6}$$

where the value of σ changes randomly from 1 to 0 in every iteration. The motion produces similar patterns as those observed in fish schooling[40] and gazelle-lion[41] models.

Swarm $\mathcal{H}$ can show two features using the two above-defined objectives. First, swarm $\mathcal{H}$ can remain in the surveillance space or can escape to its base station as the second feature. We assume that h_j UAVs' base station is placed outside of the surveillance area, and they can leave the surveillance area rapidly. The new objective does reduce the distance of each h_j from the destination point (base station). Combining this objective with the above-mentioned objectives, h_j UAVs show zig-zag motion when they are evading to their base station. Such behavior is realized by minimizing c_3:

$$c_3 = \frac{\Gamma}{\sum_{f=1}^{n} \|p_j(t+1) - p_{bs}\|}, \tag{7}$$

where Γ is either equal to one or zero depending on the selected feature. In addition, p_{bs} is the position of base station which is located in the outside of the search area.

3. Development of Monitoring Swarm

We design searching tasks for having an effective monitoring performance, and we assume that swarm $\mathcal{M}$ dispersed well using proper dispersion algorithm. In search mode, each UAV maximizes its search area, which is a combination of two objectives: the UAV's position and the camera orientation. The position is selected in a way to maintain a suitable distance among m_i UAVs. Additionally, camera orientation is chosen considering the camera orientation of its neighbor's for having a minimum or, if possible, no overlap. Such objectives are achieved by m_i UAVs by solving a multi-objective problem, which is formulated in the next section.

3.1. *Development of search objectives*

For the monitoring swarm of UAVs, the search approach is developed according to the method in ref. [35]. In this approach, each UAV $m_i \in \mathcal{M}$ plans its next position and camera FOV for time $t+1$, which is represented by a corresponding vector:

$$s_i(t+1) = (p_i(t+1), f_i(t+1)), \tag{8}$$

where $i = 1, 2, \ldots, I$.

The main goal considered for the entire swarm is to maximize the collective patrolling coverage at every step. To attain such a global goal by controlling each UAV locally, the objectives of each UAV are to avoid the same camera coverage of its neighbor and keep a minimum distance from its peer. Specifically, subject to constraint Eqs. (1)–(3) and using the states of all known neighbors within the communication range, the first objective for UAV i is given by:

$$o_1 = \sum_{k=1}^{n} \|f_i(t+1) - f_k(t)\|, \tag{9}$$

where $f_i(t+1)$ represents the next camera orientation and $f_k(t)$, $k =, 1, 2, \ldots, n$ represent the current camera orientations of all the neighbors. By maximizing objective o_1, each UAV tries to have different camera orientations than the neighbors to have larger collective camera coverage.

The second objective is to maintain a minimum distance from the other peers while dispersing in the area, which is given by minimizing the following objective function:

$$o_2 = \sum_{k=1}^{n} l\Delta p_k \, exp(\eta \Delta p_k), \tag{10}$$

where Δp_k is the Euclidean distance of the next position from the current position of the UAV, which is defined as:

$$\Delta p_k = \|p_i(t+1) - p_k(t)\|, \tag{11}$$

and parameter l is dynamically tuned as

$$l = \begin{cases} a \text{ if } \dfrac{R_{\max}}{8} < \Delta p_k \leq \dfrac{R_{\max}}{6} \\ b \text{ if } \dfrac{R_{\max}}{10} < \Delta p_k \leq \dfrac{R_{\max}}{8} \\ c \text{ if } \Delta p_k \leq \dfrac{R_{\max}}{10} \end{cases} \tag{12}$$

where

$$R_{\max} = p_i + R. \tag{13}$$

Using o_2, each UAV is maximizing the distance with other peers, while Eq. (12) penalizes UAVs that are too close with other peers. The rationale behind the proposed objectives (O_1,O_2) is that the proper positioning and proper setting of the camera orientation of searchers maximize the patrolling coverage that improves search efficiency. We assume all UAVs have preliminary information about the search location, and each UAV deploys itself initially. According to the above objective, they try to be dispersed by reasonable distance from each other.

Patrolling is a very complex task as the individual UAVs have to take cooperative moves continuously in the environment without any global coordination. A simple single-objective optimization approach cannot make a balanced move of UAVs by trading off various aspects of such a complex scenario. On the contrary, the MOO approach is capable of deciding the proper move with contradictory objectives of the UAVs, and consequently, the swarm is expected to perform more efficiently in attaining the collective goal of detecting intruders. Only two objectives are used to manage the optimization burden reasonable while providing robust behavior of UAVs in the dynamic environment. In the next section, we develop different search solutions using the above-mentioned objectives.

3.2. Solution techniques

Using the objectives described above, the performance of the patrolling swarm is evaluated using the following three different implementation techniques to identify the best one.

3.2.1. Single swarm implementation. This method simply releases the patrolling swarm to the area of interest for searching. The swarm searches the entire surveillance space without restricting

https://doi.org/10.1017/S0263574721000059 Published online by Cambridge University Press

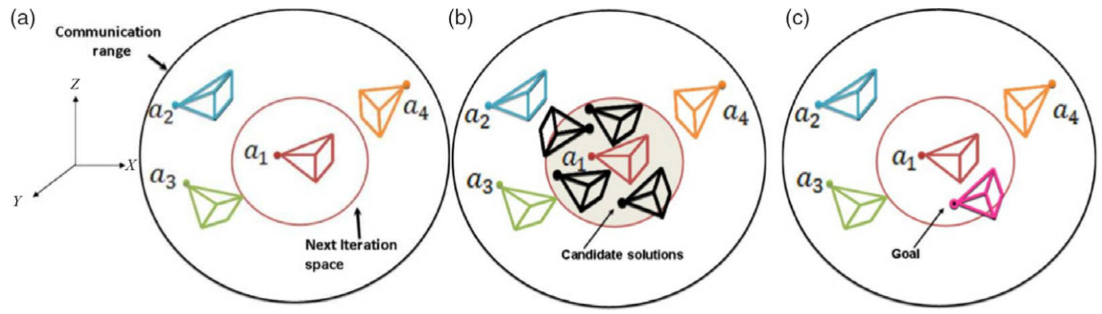

Fig. 1. The process of planning the next position for UAV a_1 considering its peers (a_2,a_3,a_4) within communication range: (a) the status at time t, (b) a way of generating candidate solutions for next iteration, and (c) choosing the best solution.

movements of its members within some sub-space, that is, individual UAVs freely move throughout the entire space.

3.2.2. Multi-swarm implementation. This method divides the space into some sub-spaces of equal size and allocates all swarm members into the specific sub-spaces to form sub-swarm, that is, members' movements are restricted within the sub-space. Each sub-swarm has an equal number of members and they stay within the sub-space for the entire observation period.

3.2.3. Mapping. This method divides the entire surveillance space into cells (N_{cell}) and creates a grid map to keep the visiting time trace of swarm members. The approach imitates the pheromone-based communication mechanism found in the ant colony system.[18] The grid map contains decaying pheromone markers to indicate the last time of visiting by a member. Therefore, any cells that have not been visited by any member recently can be identified easily. It is assumed that each swarm member has access to information of adjacent cells from the grid map. In practice, the update of the grid map and sharing information can be realized using the gossip-based approaches using ad-hoc communication,[19] where members share their search history with the other peers. Using such shared information, a single swarm is used to search the entire space.

The procedure after building a map is as follows: first, the search space is divided into S sub-spaces, and then the last search time for each sub-space is computed using the current time and the last searched time. Then closest UAVs are targeted to specific unsearched regions that the time of the last search exceeds the predefined threshold time (T_t). In this paper, the whole space is divided into N_{cell}.

3.3. Implementation of MOO

The proposed MOO is implemented in a discrete-time framework using an embedded computing system that determines the next pose of each UAV at each time step t. In this technique, several pose offsets are randomly generated in a virtual sphere of radius Rv centering the UAV's current, as shown in Fig. 1. Using each of these candidate solutions, objective functions are evaluated to select the best solution. The UAV then moves to the target pose at the next step t+1.

Updates of the pose of UAV h_j are similar to Eq. (5).

For distribution of F candidates inside the virtual sphere around the m_i UAV (as a center of the virtual sphere), we have

$$p_g = p_c - R_v + (p_c + R_v - (p_c - R_v)) \times \delta, \tag{14}$$

where δ is a random number in the range of (0,1]. In addition, p_g and p_c are generated position and the current position of the m_i UAV. Furthermore, R_v is the virtual sphere's radius, and rand is a unit random variable. R_v is the radius of the virtual spheres when the area of interest is bounded.

Both m_i and h_j UAVs use different multi-objectives, namely multi-objective evolutionary algorithm using decomposition (MOEA/D) and multiple-objective particle swarm optimization (MPSO). We choose these heuristic optimization algorithms according to our earlier work,[35] which shows both MPSO and MOEA/D perform well in the search problem.

https://doi.org/10.1017/S0263574721000059 Published online by Cambridge University Press

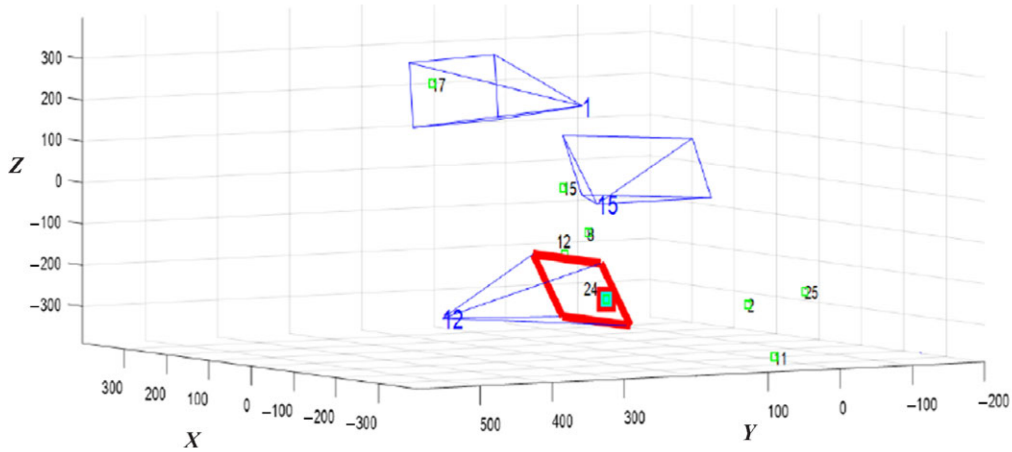

Fig. 2. The simulation environment: searcher ID 12 has detected intruder ID 24.

3.3.1 The procedure of the proposed MOO system

Step 1: Receive the user's preferences and initialize the algorithm parameters.
Step 2: Position the h_j UAVs in set mode.
Step 3: Run the user's preferred search techniques. Perform positioning and camera alignment for m_i UAVs. (Start the MOO, finding a set of non-dominated solutions, and choose the best solution according to a decision-maker).
Step 4: Go to step 2 until stopping criteria (maximum iterations) is satisfied.

4. Experiments and Results

We develop a swarm simulator using MathWorks Matlab running on a desktop PC with an Intel Core i7 processor and 16GB RAM. For m_i UAVs, the HFOV and VFOV are 21.77×21.77 degrees to a depth of 200 m as a DFOV, and the communication range is 1000 m, which with growing off-the-shelf hardware can be implemented in practice.[42] For h_j UAVs, the FOV is 360 degree with a finite depth of 300 m (e.g., Ricoh Theta[4]), and also with a limited communication range of 1000 m. The maximum speeds for m_i and h_j UAVs are set to $V^M_{\max} = 40$ m/s and $V^H_{\max} = 35$ m/s, respectively. In addition, we defined two different space sizes (Γ_S), namely smaller space (Γ_1) and larger space (Γ_2), which are set to $500 \times 500 \times 500$ m^3 and $1000 \times 1000 \times 1000$ m^3, respectively. In this paper, we arbitrarily choose environments with two different volumes to observe how the environment size influences searchers' performance. In selecting the sizes of the environment, it is ensured that the camera view remains much smaller than the environment size to keep the searching task challenging enough for evaluating the proposed method. An image of the simulation environment is shown in Fig. 2 where a searcher has detected an intruder. As described in the earlier section, MOEA/D is applied for m_i UAVs, and MPSO is applied for h_j UAV. In both MOEA/D and MPSO algorithms, the population size is set to 15, and results are obtained after running it for 50 generations. The population size of 15 is chosen in this study according to some sensitivity analysis. Specifically, we have applied different ranges of population sizes, and it is found that the population size lower than 15 reduces the performance while a higher population size does not have a distinctive impact on the obtained results. Each simulation lasts for a specific time, which is equal to 200 iterations. Other related details are described in Tables I and II. The heuristic optimization has stochastic nature, therefore for a statistically meaningful comparison, all results of this section are taken from the average of 30 independent runs of the respective simulation using the same initial settings for both swarms.

The proposed framework of monitoring swarm using UAVs with above settings is evaluated considering various aspects. Specifically, the comparison of the algorithms and methods is conducted first, which are followed by the study on various configurations and their impact on the overall swarm performance.

[4]https://theta360.com/uk/.

https://doi.org/10.1017/S0263574721000059 Published online by Cambridge University Press

Table I. Implementation-specific parameters.

Par	**Value**	**Par**	**Value**	**Par**	**Value**
R_v	500 m	r	100 m	σ	0.1–1
R	50 m	$C_{\max}$	3	n_1	5
a	5	b	6	c	8
Ref	[0.0207, 0.016]	N_{cell}	8	α	200
β	200	γ	200	η	1

Table II. The parameters of MOOs.

Algorithm	**Parameter**	**Value**
MPSO[43]	Max archive size	10
	c_1	1
	c_2	2
	Grid inflation (α)	10
	Leader selection pressure (β)	2
	Number of grids per each dimension	7
MOEA/D[44]	Max archive size	20
	Subproblems	3
	Number of neighbors (T)	3
	Mutation rate	0.5

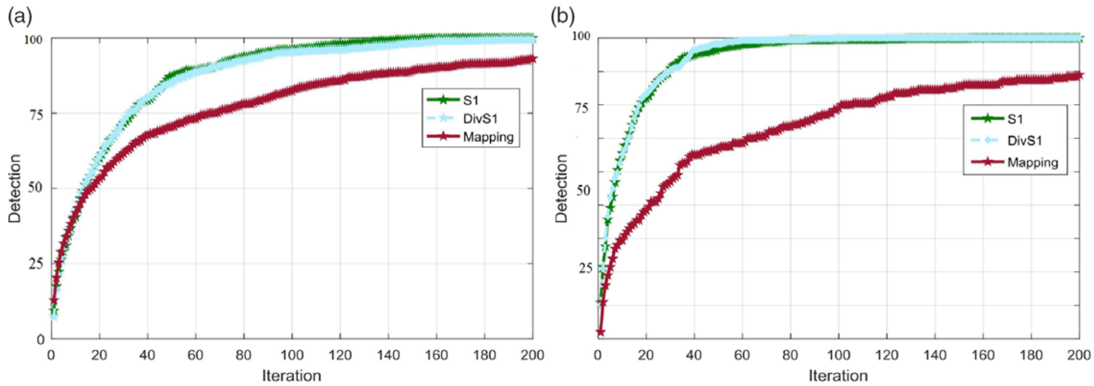

Fig. 3. Comparison of the average detection percentage among various strategies when the number of m_i UAV is (a) 16 and (b) 32.

4.1. *Evaluation of search strategies*

In this evaluation, we consider 20 h_j UAVs that are randomly placed in the environment and they remain in the surveillance space for the entire simulation period. We introduce the monitoring swarm to detect all h_j UAVs and observe the success of the swarm in terms of the detection rate over simulation time. Figure 3 shows the progressive performance, in terms of detection percentage, of the monitoring swarm with 16 and 32 members, respectively. In each case, three search approaches are compared, which are single swarm search (S1), mapping (Mapping), and sub-swarming (DivS1).

In both cases of 16 and 32 UAVs in the monitoring swarm, the mapping approach, which splits the entire space into small cells and keeps a record of their cell visits, could slowly detect h_j UAVs as shown in Fig. 3. Interestingly, the swarm search, which employs a single swarm without splitting the space or assigning individuals to a specific sub-space, works similar to sub-swarming, which divides the surveillance area into sub-space and assigns a few UAVs or a small swarm to each sub-space. Both S1 and DivS1 outperform mapping in terms of both the progressive and final detection rates. Between these two, swarming has less complexity in practice comparing to sub-swarming, where it requires proper scattering of UAVs in the predefined cells. This implies that the collective search approach is the best since it wisely sends its UAVs towards the dense area of h_j nodes, whereas the mapping approach often keeps its individual to an unimportant sub-space even when no or very few

https://doi.org/10.1017/S0263574721000059 Published online by Cambridge University Press

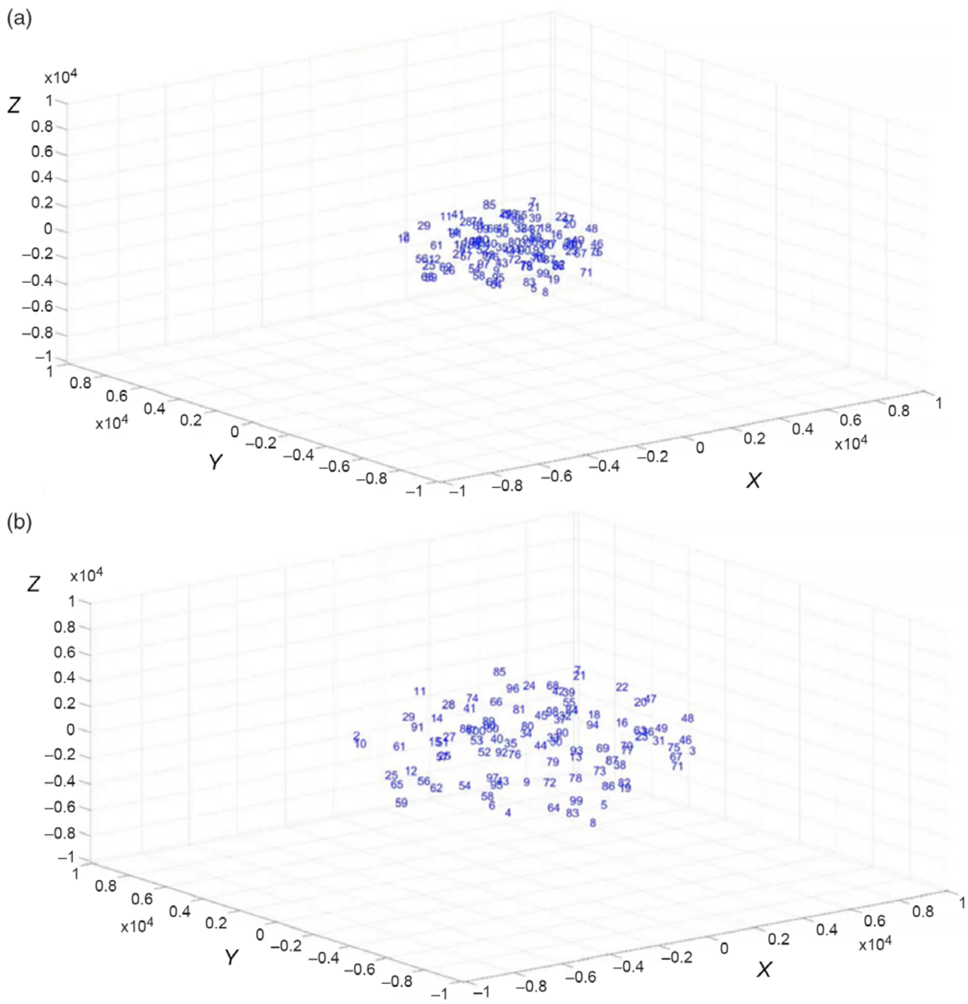

Fig. 4. Two snapshots from the simulator between iterations (a) 1 and (b) 37.

h_j UAVs exist. Another drawback of the sub-swarming approach is that after detecting of all h_j UAVs in the sub-space, the corresponding m_i UAVs remains in the area. This is clearer when the h_j UAVs leave the surveillance area rapidly as shown in Fig. 5. On the other hand, in the swarming search approach after each detection, a monitoring UAV may move to the other part of the surveillance area for further detection and contributes to the global achievement of the swarm. These impressive search results illustrate the capability of the proposed method and formulation of the framework. The finding of this simulation also confirms findings of refs. [45, 46] where single swarm outperformed other swarming techniques. Two snapshots of simulation take at iteration 1 and 37 are shown in Fig. 4.

In addition to the above progressive evaluation of the three strategies, we have also evaluated their overall detection performance in a very special case, where some of the h_j UAVs exist in the surveillance area for a short time. The proposed approaches are tested in terms of total detections when h_j UAVs escape from the surveillance area to the outside. In this sense, the searchers have a very limited time to find the h_j UAVs, and it is a good measure to see how the different approaches can deal with this realistic issue. According to Fig. 5, S1 and DivS1 perform better than Mapping. It should be noted that in the case of evasive h_j UAVs, their fast detection is crucial. In fact, a major advantage of S1 is global efficient coverage.

The accuracy of the algorithm varies depending on the number of searchers, their sensor ranges, and searching time. Furthermore, the accuracy of the search process also depends on the intruders™ hiding strategies, the number of intruders, and the surveillance space size.

Hereafter, according to the above, S1 is chosen as the best search strategy in the following sections. Next, this search strategy, S1, is compared with an existing approach called Levy flight, which

https://doi.org/10.1017/S0263574721000059 Published online by Cambridge University Press

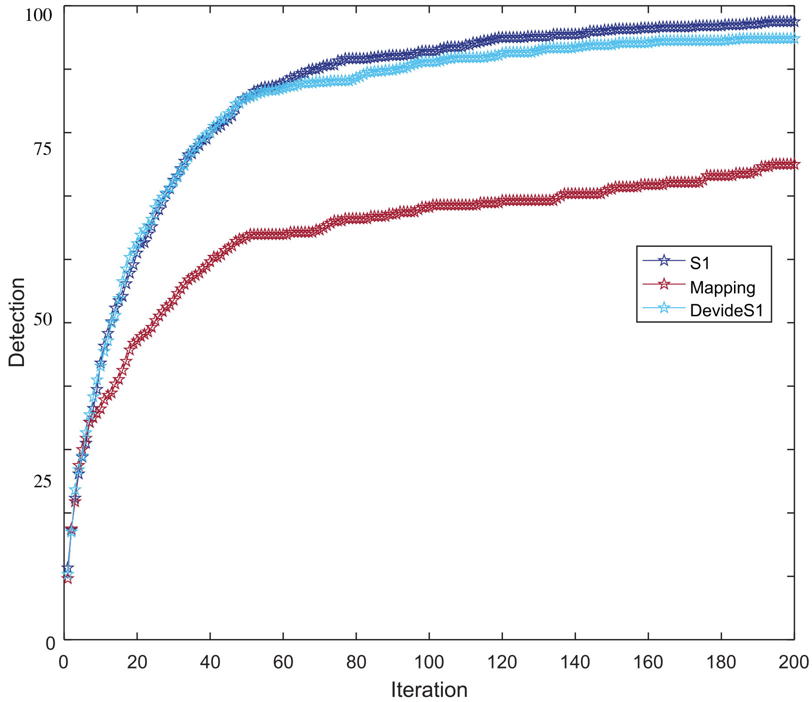

Fig. 5. The comparison among the approaches for different number of m_i UAVs when h_j UAVs are escaping from the simulation environment to the outside after iteration 50.

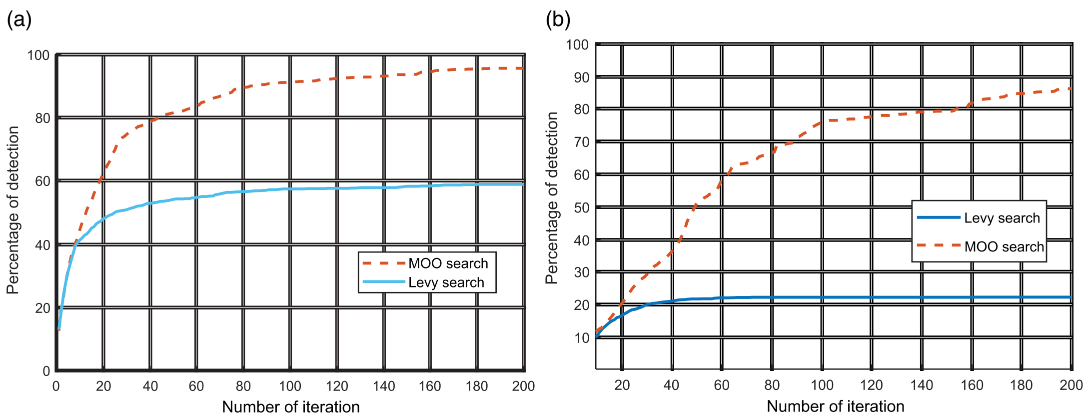

Fig. 6. The comparison between MOO search and traditional Levy flight search. (a) Environment I (smaller) (1) and (b) environment II (larger).

is a well-known traditional search approach. This search method generates a random jump for each searcher. In Fig. 6(a), the search performance is compared when there are 16 searchers and 40 intruders in the smaller environment (Γ_1) . We want to evaluate the collective search performance when the number of intruders is much higher than the searchers and observe whether searchers are perplexed by detecting multiple intruders concurrently. Considering 16 searchers in the environment, we choose 40 intruders that are more than double the searchers' number. The comparison in this figure illustrates the effectiveness of S1 comparing to Levy flight. Since search has a higher complexity in larger environments, in Fig. 6(b), it is found that MOO outperforms the traditional Levy flight.

4.2. Impact of hardware configuration

The purpose of this evaluation is to determine how certain hardware configuration impacts the search performance. The studied configuration is limited to the communication range and the maximum velocity.

https://doi.org/10.1017/S0263574721000059 Published online by Cambridge University Press

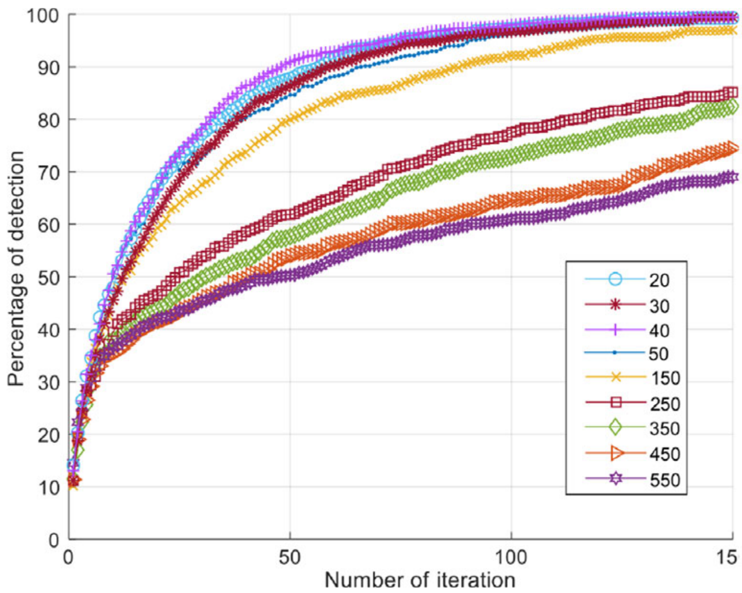

Fig. 7. Effects of communication range on the detection rate of m_i UAVs.

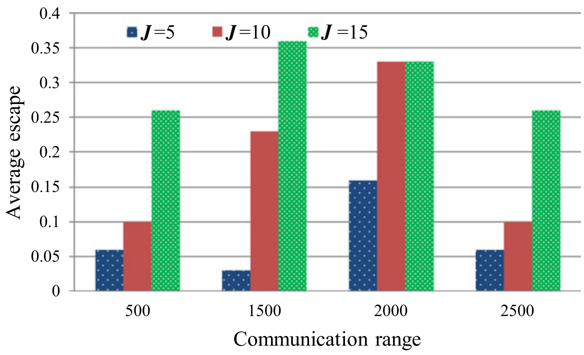

Fig. 8. The comparison of communication ranges in h_j UAVs when m_i UAVs work with multi-swarm implementation, and h_j UAVs vary from 5 to 15 with different communication range.

4.2.1. Communication range. The main goal of this experiment is to study the impact of different communication ranges in detection rates. According to Fig. 7, a higher communication range in S1 among m_i UAVs does not always result in better performance. In other words, a communication range in the patrolling swarm does not always indicate an enhanced performance. According to the simulation result, after a point, a higher communication range leads to a drop in the performance. This is due to the fact that a very high communication range leads to adding too much information to searchers. This distracts searchers with unimportant data that result in improper functioning.

Since limited FOV in patrolling swarm grows the stochastic results in detections of intruders, it is difficult to interpret how the communication range in intruders may affect patrolling swarm's total detections. To come up with this, a FOV of the patrolling swarm in the evaluation of communication range of h_j is set to 360°. As a result, in this only specific evaluation, a patrolling swarm can detect any intruders within its camera range. In addition, to get more distinctive results, in this experiment, the environment is set to a larger environment (Γ_2) while there are 16 patrolling UAVs and 40 intruders. According to Fig. 8, in zig-zag trajectory mode, the h_j UAVs communication range could slightly affect the number of the escaped h_j UAVs. The trend in Fig. 9 shows with increasing communication range of h_j UAVs, until a specific point, a number of non-detected UAVs also increase. According to results, larger communication ranges lead to better escaping behavior until a certain point; this is due to the fact that with higher communication range h_j UAVs are more aware of m_i UAVs which help intruders to avoid being detected by patrolling UAVs.

https://doi.org/10.1017/S0263574721000059 Published online by Cambridge University Press

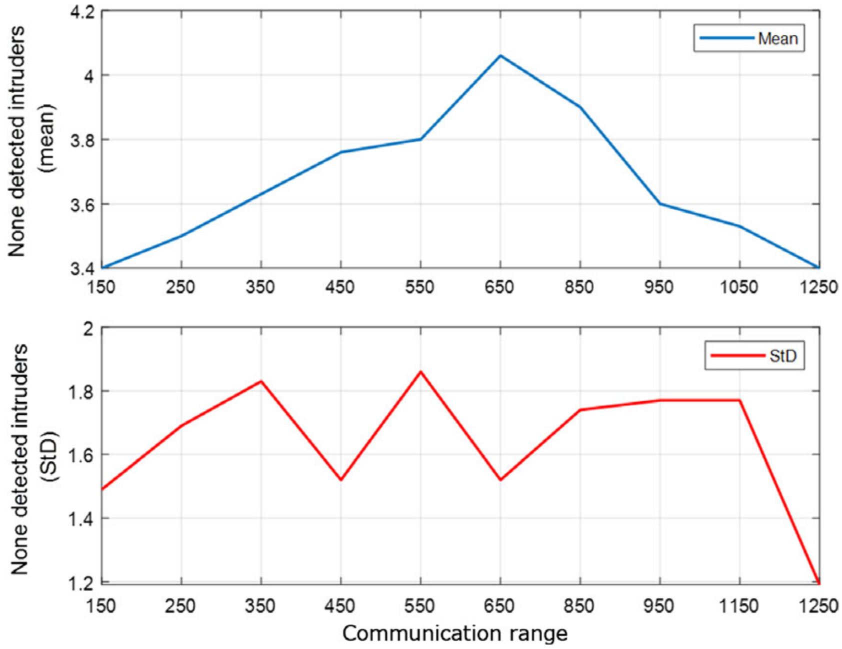

Fig. 9. Communication range of h_j UAVs' effect on detection rate of m_i UAVs.

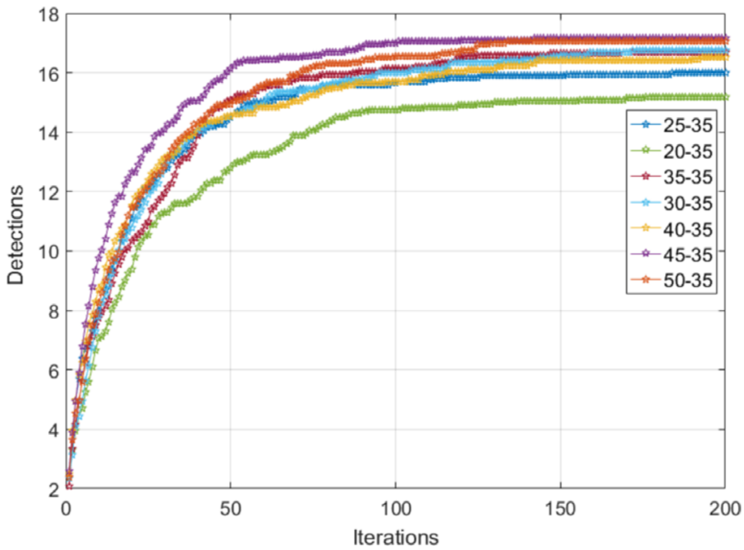

Fig. 10. The impact of velocity ratio between the m_i searchers and h_j intruders in detections of h_j UAVs where h_j has a fixed velocity (35 m/s).

In short, higher communication range improves the performance and leads to higher awareness about the area in intruding UAVs. After reaching a certain point, the increase of communication range deteriorates the performance of intruders.

4.2.2. Velocity impact on detections. Velocity relations between the searchers and evaders affect the rate of detections. The evaders evade from the area of interest to the outside as shown in Fig. 10. We compare different relations between the maximum speed in UAVs in swarm $\mathcal{M}$ and swarm $\mathcal{H}$. As the speed of h_j UAVs going larger than m_i UAVs, the successful detections in m_i UAVs fall, and similarly, when m_i UAVs have higher speed comparing to h_j UAVs there is higher detection.

https://doi.org/10.1017/S0263574721000059 Published online by Cambridge University Press

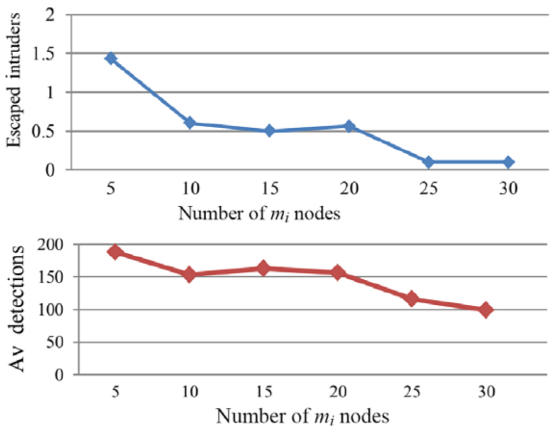

Fig. 11. The impact of number of m_i UAVs in successful escape of h_j UAVs and average iteration for all detections.

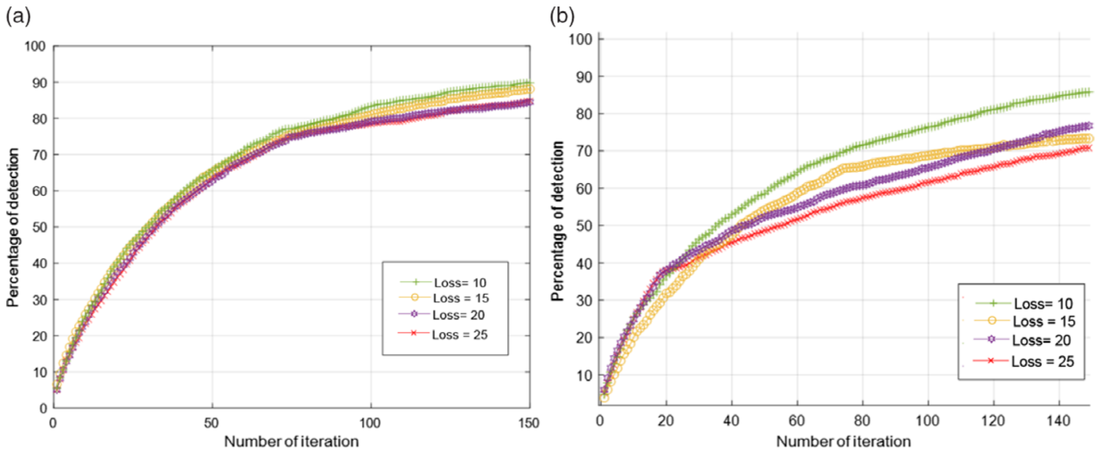

Fig. 12. The impact of abrupt failure of m_i UAVs at iteration (a) 30 and (b) 75.

4.3. *Other specifications*

A few specifications are common in swarm robotics, for example, effect of failure in the members or effect of the number of swarm members. We conducted a few experiments to study some other issues.

4.3.1. Number of m_i UAVs. To show how the number of UAVs affects the search in terms of detections. As shown in Fig. 11, as the number of m_i UAVs increases, the number of escaped h_j UAVs decreases. In addition, the average iteration of all detections by m_i UAVs decreases as a result of increased number of m_i UAVs. This trend can be generalized to all settings, and in this experiment, h_j UAVs are set to 10 UAVs when m_i UAVs vary from 5 to 30.

4.3.2. Abrupt fails or removal of m_i UAVs. In this test, the robustness of the search algorithm to the failure of m_i UAVs due to malfunction or any other reason is studied. In this specific test, the whole simulation time is set to 150 iterations, and the abrupt failure is happened in both initial and final iterations to test the algorithm.

In the first experiment, a certain number of m_i UAVs (10, 15, 20, and 25) failed at iteration 30 suddenly. As shown in Fig. 12(a), there is a slight difference in performance regarding total detections of h_j nodes.

In another experiment, leaving of searchers in the half of total studied iterations (iteration 75 out of 150) is studied. The simulation results in both experiments confirm the robustness of the algorithm in such situations (Fig. 12(b)).

https://doi.org/10.1017/S0263574721000059 Published online by Cambridge University Press

5. Conclusion

This paper has developed a swarming search technique for patrolling UAVs to efficiently detect moving objects or intruding UAVs in a 3D space. Specifically, we have designed a MOO approach for searching that incorporates several strategies to improve the search quality in the swarm versus swarm context. Several hardware configurations of both the intruding and patrolling UAVs, in terms of communication ranges, maximum velocity, and camera coverage, are evaluated by observing their detection and escaping performances. The evaluation results indicate that a single swarm is the best choice when the intruding swarm stays or leaves the surveillance space. Next, the simulation results reflect that the high communication range does not always indicate a swarm's better performances. The findings show that the communication range in the patrolling swarm after reaching a threshold deteriorates the performance, while a higher communication range until a certain point in the intruding swarm improves the escaping performance. The patrolling swarm performances can be improved further by modeling the intruders' behavior and tackling them accordingly, which we keep as future work.

References

1. G. Cai, J. Dias and L. Seneviratne, "A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends," *Unmanned Syst.* **2**(02), 175–199 (2014).
2. C. Stöcker, R. Bennett, F. Nex, M. Gerke and J. Zevenbergen, "Review of the current state of UAV regulations," *Remote Sens.* **9**(5), 459 (2017).
3. D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature* **521**(7553), 460 (2015).
4. R. L. Finn and D. Wright, "Unmanned aircraft systems: Surveillance, ethics and privacy in civil applications," *Comput. Law Secur. Rev.* **28**(2), 184–194 (2012).
5. C. Schlag, "The new privacy battle: How the expanding use of drones continues to erode our concept of privacy and privacy rights," *Pitt. J. Tech. L. Pol'y* **13**(2), 1–27 (2012).
6. N. McDermott, "Police drones are grounded ... for breaking the law edition 3," Feb 16 2010. Copyright - Copyright (c) Associated Newspapers Ltd. 2010; Last updated - 2012-10-25.
7. Y. Fu, M. Ding and C. Zhou, "Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV," *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **42**(2), 511–526 (2012).
8. V. San Juan, M. Santos and J. M. Andújar, "Intelligent UAV map generation and discrete path planning for search and rescue operations," *Complexity*, **2018**, 1–17 (2018).
9. C. Gomez and H. Purdie, "UAV-based photogrammetry and geocomputing for hazards and disaster risk monitoring–a review," *Geoenviron. Disasters* **3**(1), 23, (2016).
10. M. A. Kovacina, D. Palmer, G. Yang and R. Vaidyanathan, "Multi-Agent Control Algorithms for Chemical Cloud Detection and Mapping Using Unmanned Air Vehicles," *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002*, vol. 3 (2002) pp. 2782–2788.
11. C. Koparan, A. B. Koc, C. V. Privette, C. B. Sawyer and J. L. Sharp, "Evaluation of a UAV-assisted autonomous water sampling," *Water* **10**(5), 655 (2018).
12. D. Mader, R. Blaskow, P. Westfeld and C. Weller, "Potential of UAV-based laser scanner and multispectral camera data in building inspection," *Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci.* **41**, 1135–1142 (2016).
13. M.-R. Andervazh, J. Olamaei and M.-R. Haghifam, "Adaptive multi-objective distribution network reconfiguration using multi-objective discrete particles swarm optimisation algorithm and graph theory," *IET Gener. Transm. Distrib.* **7**(12), 1367–1382 (2013).
14. L. D. Stone, R. L. Streit, T. L. Corwin and K. L. Bell, *Bayesian Multiple Target Tracking* (Artech House, USA, 2013).
15. Y. Altshuler, A. Pentland and A. M. Bruckstein, "The Cooperative Hunters–Efficient and Scalable Drones Swarm for Multiple Targets Detection," **In:** *Swarms and Network Intelligence in Search* (Springer, 2018) pp. 187–205.
16. S. M. D. M. Senanayake, Tracking of Large Crowds with a Swarm of Aerial Robots *Master Thesis* (Monash University, 2017).
17. T. Lochmatter, X. Raemy and A. Martinoli, Odor Source Localization with Mobile Robots, Tech. Rep. (2007).
18. S.-H. Oh and J. Suk, "Evolutionary controller design for area search using multiple UAVs with minimum altitude maneuver," *J. Mech. Sci. Tech.* **27**(2), 541–548 (2013).
19. C. Erignac, "An Exhaustive Swarming Search Strategy Based on Distributed Pheromone Maps," *AIAA Infotech@ Aerospace 2007 Conference and Exhibit* (2007) p. 2822.
20. A. T. Hayes, A. Martinoli and R. M. Goodman, "Swarm robotic odor localization: Off-line optimization and validation with real robots," *Robotica* **21**(4), 427–441 (2003).
21. N. R. Hoff, A. Sagoff, R. J. Wood and R. Nagpal, "Two Foraging Algorithms for Robot Swarms Using Only Local Communication," *2010 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (IEEE, 2010) pp. 123–130.

https://doi.org/10.1017/S0263574721000059 Published online by Cambridge University Press

22. H. Lau, S. Huang and G. Dissanayake, "Optimal Search for Multiple Targets in a Built Environment," *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE Press, 2005).
23. T. H. Chung, G. A. Hollinger and V. Isler, "Search and pursuit-evasion in mobile robotics," *Auto. Robots* **31**(4), 299 (2011).
24. L. Gregorin, S. N. Givigi, E. Freire, E. Carvalho and L. Molina, "Heuristics for the multi-robot worst-case pursuit-evasion problem," *IEEE Access* **5**, 17552–17566 (2017).
25. J. P. How, C. Fraser, K. C. Kulling and L. F. Bertuccelli, "Increasing autonomy of UAVs," *IEEE Robot. Autom. Mag.* **16**(2), 43–51 (2009).
26. K. Wray and B. Thompson, "An Application of Multiagent Learning in Highly Dynamic Environments," *AAAI Workshop on Multiagent Interaction Without Prior Coordination (MIPC 2014)* (2014).
27. S. Changhai, L. Ding and D. Xiaobo, "Dynamic programming algorithm for the detection of air dim target," *IET International Radar Conference 2013*, (2013) pp. 1–3, doi: 10.1049/cp.2013.0250.
28. S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace Electron. Syst. Mag.* **19**(1), 5–18 (2004).
29. H. Yu, K. Meier, M. Argyle and R. W. Beard, "Cooperative path planning for target tracking in urban environments using unmanned air and ground vehicles," *IEEE/ASME Trans. Mech.* **20**(2), 541–552 (2015).
30. S. M. Tonissen and R. J. Evans, "Performance of dynamic programming techniques for track-before-detect," *IEEE Trans. Aerospace Electron. Syst.* **32**(4), 1440–1451 (1996).
31. A. Alexopoulos, T. Schmidt and E. Badreddin, "Cooperative Pursue in Pursuit-Evasion Games with Unmanned Aerial Vehicles," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2015) pp. 4538–4543.
32. T. H. Chung, M. R. Clement, M. A. Day, K. D. Jones, D. Davis and M. Jones, "Live-Fly, Large-Scale Field Experimentation for Large Numbers of Fixed-Wing UAVs," *2016 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2016) pp. 1255–1262.
33. D. T. Davis, T. H. Chung, M. R. Clement and M. A. Day, "Multi-Swarm Infrastructure for Swarm versus Swarm Experimentation," *Distributed Autonomous Robotic Systems* (Springer, 2018) pp. 649–663.
34. R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Control* **51**(3), 401–420 (2006).
35. A. Moltajaei Farid, S. Egerton, J. C. Barca and M. A. S. Kamal, "Adaptive Multi-Objective Search in a Swarm vs Swarm Context," *Proceedings of IEEE Conference on Systems, Man, and Cybernetics* (2018).
36. J. A. Preiss, W. Honig, G. S. Sukhatme and N. Ayanian, "Crazyswarm: A Large Nano-quadcopter Swarm," *2017 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2017) pp. 3299–3304.
37. T. H. Chung, K. D. Jones, M. A. Day, M. Jones and M. Clement, "50 vs. 50 by 2015: Swarm vs. swarm UAV live-fly competition at the naval postgraduate school," *The NPS Institutional Archive*, (2013) pp. 1792–1811.
38. M. Senanayake, I. Senthooran, J. C. Barca, H. Chung, J. Kamruzzaman and M. Murshed, "Search and tracking algorithms for swarms of robots: A survey," *Robot. Auto. Syst.* **75**, 422–434 (2016).
39. H. Wang, M. Olhofer and Y. Jin, "A mini-review on preference modeling and articulation in multi-objective optimization: Current status and challenges," *Complex Intell. Syst.* **3**(4), 233–245, (2017).
40. S. Luke and L. Spector, "Evolving Teamwork and Coordination with Genetic Programming," *Proceedings of The 1st Annual Conference on Genetic Programming* (MIT Press, 1996) pp. 150–156.
41. V. Gómez, S. Thijssen, A. Symington, S. Hailes and H. J. Kappen, "Real-Time Stochastic Optimal Control for Multi-Agent Quadrotor Systems," *ICAPS* (2016) pp. 468–476.
42. N. Muchiri, S. I. Kamau and B. W. Ikua, "Architectures and Algorithms for Multiple UAV Cooperative Control: A Review," *Proceedings of Sustainable Research and Innovation Conference* (2018) pp. 180–183.
43. C. A. C. Coello, G. T. Pulido and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.* **8**(3), 256–279 (2004).
44. Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.* **11**(6), 712–731 (2007).
45. N. Chmait, D. L. Dowe, Y.-F. Li, D. G. Green and J. Insa-Cabrera, "Factors of Collective Intelligence: How Smart are Agent Collectives?," *ECAI* (2016) pp. 542–550.
46. N. Chmait, Y.-F. Li, D. L. Dowe and D. G. Green, "A Dynamic Intelligence Test Framework for Evaluating AI Agents," *Proceedings of the Workshop Evaluating General-Purpose AI, EGPAI* (2016) pp. 1–8.
47. R. Cheng, Y. Jin, M. Olhofer and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.* **20**(5), 773–791 (2016).

Appendix

In this section, we discuss the decision-making process to choose the best solution from the Pareto front. Firstly, in the decision-maker setup, the number of solutions in Pareto front varies in each iteration. Vector-based approach can work with any number of solutions in the Pareto front, so the vector-based decision-maker is applied when the vector passes from a reference point (*Ref*).

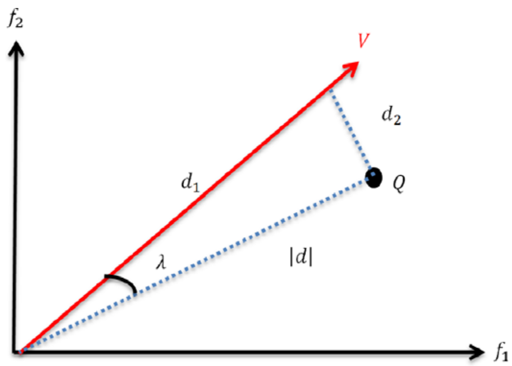

Fig. A.1. Vector-based preference decision-making with two objectives (f_1 and f_2) where Q is a solution that we wish to compute its preference value using the reference vector V.

Reference vectors provide a preference of the designer in the solution space. Each solution is compared with the perpendicular distance d_1, while d_2 improves the diversity as shown in Fig. A.1.[47] To do this, the following equation computes the preference-based intersection value.

$$g_1 = d_1 + \lambda d_2, \tag{A1}$$

where g is preference value, and the smallest value of g is the most preferred solution. λ is

$$\lambda = \cos^{-1}\left(\frac{\vec{V}.\vec{P}}{\|\vec{V}\|\|\vec{P}\|}\right), \tag{A2}$$

$\vec{V}$ is a reference vector connecting the center to Ref, $\vec{P}$ is a vector connecting the center to point Q, and $|\,.\,|$ represents dot product. The point Q represents a solution on Pareto front that we would like to compute its preference value.[39]

Secondly, for the scenario that we have more than two objectives, the vector-based decision-maker could not perform well. Therefore, we have used a new decision-maker. The idea behind this decision-maker is quite similar to the above-mentioned vector-based decision-maker.[47] The preference value is as follows:

$$g_2 = (1 + p(\lambda))\|d\|, \tag{A3}$$

where $p(\lambda)$ is a penalty function and is similar to:[47]

$$p(\lambda) = O\frac{\lambda}{\gamma}, \tag{A4}$$

where γ is the minimum angle between $\vec{V}$ and other $\vec{P}$ for all objectives, and O is the total number of objectives.