Design and simulation of the emergent behavior of small drones swarming for distributed target localization

Antonio L. Alfeo[1,2], Mario G. C. A. Cimino[1], Nicoletta De Francesco[1], Massimiliano Lega[3], Gigliola Vaglini[1]

Abstract

A swarm of autonomous drones with self-coordination and environment adaptation can offer a robust, scalable and flexible manner to localize objects in an unexplored, dangerous or unstructured environment. We design a novel coordination algorithm combining three biologically-inspired processes: stigmergy, flocking and evolution. Stigmergy, a form of coordination exhibited by social insects, is exploited to attract drones in areas with potential targets. Flocking enables efficient cooperation between flock mates upon target detection, while keeping an effective scan. The two mechanisms can interoperate if their structural parameters are correctly tuned for a given scenario. Differential evolution adapts the swarm coordination according to environmental conditions. The performance of the proposed algorithm is examined with synthetic and real-world scenarios.

Keywords: Swarm Intelligence, Drone, Stigmergy, Flocking, Differential Evolution, Target search

*Corresponding authors: Antonio L. Alfeo, Mario G. C. A. Cimino
Email addresses: `luca.alfeo@ing.unipi.it` (Antonio L. Alfeo), `mario.cimino@unipi.it` (Mario G. C. A. Cimino)
[1]University of Pisa, largo Lucio Lazzarino 1, Pisa, 56122, Italy
[2]University of Florence, via di Santa Marta 3, Florence, 50139, Italy
[3]University of Naples "Parthenope", Centro Direzionale Isola C4, Naples, 80143, Italy

1. Introduction and Motivation

1.1. Motivation

In this paper we consider the problem of discovering static targets in an unstructured environment, employing a set of small dedicated Unmanned Aircraft Vehicles (UAVs) and minimizing the total time spent to discover targets. The main motivation for this study comes from the request to deal with circumstances where the target and the space of exploration are poorly specified, and the coordination strategy is autonomous, robust, resilient, and adaptive. Indeed, the current UAVs hardware and the available flight control software can offer good solutions to problems in many fields. However, the software available for coordinating the exploration of UAVs swarms is not sufficiently mature: limited flexibility, complex management and application-dependent design are the main issues to solve. In this paper we address the issues proposing a swarm coordination algorithm that is adaptive to different circumstances, combining three biologically-inspired processes: Stigmergy, Flocking and Evolution (SFE). For the sake of readability, each process is first introduced by using non-technical notations (ontology diagrams and graphical illustrations), and then formally specified by using mathematical and algorithmic expressions. Finally, the overall solution is experimented on various synthetic and real-world environments such as those for discovering illegal dumps and mines.

An A UAV, often called drone, is an aircraft without a human pilot onboard, since it is controlled from an operator on the ground managing a base station. Today the most common category of drones is the Small UAV (sUAV): according to US FAA (Federal Aviation Administration) rules, the term applies to an unmanned aircraft weighing less than 55 pounds. Recent advances in embedded technology encourage the use of small drones for the so called D3 missions, "dull, dirty and/or dangerous" e.g.: border and harbor patrol, landmine detection, search and rescue, and so on [1]. Obviously, the choice of the specific drone configuration depends on the mission requirements and benefits over existing ground-based instruments [2, 3]. For example, small drones, equipped with self-localization and sensing capabilities, can perform target search task in rural, natural and structurally complex environments, where access to humans and large drones is limited, dangerous or impossible. Search can be managed by using a single drone or groups of drones: to use a single drone is in general quite complex, and then search implies very high costs of design, construction and maintenance. Moreover, a unique and complex drone is vulnerable to a variety of hardware or software faults which are difficult to prevent. On the other hand, a number of considerations can support the use of groups of drones for target search [1]. When managing a group of drones, it is important to avoid centralized control that leads to exponential increases in communication bandwidth and software complexity [4]. Swarm intelligence methodologies can be applied to solve problems cooperatively maintaining scalability. The main inspiration comes from the observation of social animals, such as ants, bees, birds, and fishes, which exhibit a sort of collective intelligence achieving complex goals and coordination through simple rules and local interactions [5]. As shown in [6] in a self-organized coordination of a swarm of ground robots, each member of the swarm: (i) acts with a certain level of autonomy (ii) performs only local sensing and indirect communication; (iii) operates with decentralized control and without global knowledge, and (iv) leads to emergence of collective behavior. Among these properties, autonomy can significantly improve the search operations, especially in the presence of unknown environments [7], for which an *apriori* study of the best path to reach the targets is neither feasible nor convenient. For instance, it may be difficult or impossible to obtain a map to support search in a post-disaster scenario or under the canopy of densely forested areas.

1.2. Introduction to Stigmergy, Flocking and Evolution

The *computational stigmergy* is a reliable mechanism for generating autonomous swarm coordination. Stigmergy can create complex coordination patterns via a simple mechanism: an agent's action produces a mark, which in turn incites another action, which produces another mark, and so on. When coordinating a swarm of drones, a mark is realized in the form of virtualized pheromone, released in the current position while performing certain action. Such pheromones are sensed by other drones and incite their subsequent actions. [8]. In a typical scenario, when a physical sensor of a drone detects a target, the drone releases a particular amount of attractive pheromone on the location of the sensed target. Pheromone diffusion acts as an attractive potential on neighboring drones. While unknown targets are sensed, additional pheromone is released by flock members, thus enabling an incremental positive feedback up to completion of all targets in the proximity of the initial target. After a certain time the pheromone intensity cannot be reinforced, and in practice disappears. Furthermore, to reduce multiple explorations of the same zone, two additional coordination mechanisms are used: (i) olfactory habituation [9] and (ii) repulsive pheromone [10]. In essence, (i) a drone releases a repulsive pheromone where it does not sense a target, and (ii) a drone becomes unable to sense pheromone while moving in locations saturated by pheromones. The available drones are spatially organized into a number of flocks. Flocking behavior is an effect of local rules based on alignment, separation and cohesion [11]. In essence, with alignment, each drone tends to move in the same direction of nearby drones. With separation, each drone keeps a sufficient distance able to guarantee flexible exploration also in presence of obstacles. With cohesion each drone moves toward the center of mass of the flock to preserve the flock compactness. Results in [12] prove the benefits of the combined strategy of stigmergy and flocking and raise the need of an appropriate adaptation of the stigmergy and flock parameters to the specific search area. For example, an area with a high density of obstacles can be efficiently explored via small flocks. For the parameters adaptation we adopt an evolutionary computation technique, which improves the parameters with regard to a given measure of quality. Evolutionary computation has the advantage of making no assumptions about the problem being optimized, thus avoiding to bias the underlying mechanisms. It is based on a population of candidate solutions, iteratively improved via mechanisms inspired by natural evolution, such as mutation, selection, and crossover. In the literature, many evolutionary computation algorithms have been proposed, and many efforts have been devoted for qualitative and quantitative comparisons of such algorithms. As a result, researches on comparative analyses over benchmark problems, under the same solution representation and number of function evaluations, are already available. Taking into account such researches, we adopted the Differential Evolution (DE) algorithm, which is one of the most powerful stochastic real-parameter optimization algorithms [13]. In essence, DE operates through conventional operators of evolutionary computation, except that it perturbs the current population via scaled differences of randomly selected members. Compared with other popular methods such as genetic algorithm (GA) and particle swarm optimization (PSO), DE has attracted more attention for continuous optimization problems for its simplicity, efficiency, and adaptivity [14] [15].

1.3. Paper Structure

The paper is structured as follows. Section 2 defines the target search problem and covers a literature review of the scientific works addressing target search with swarm of UAVs. In Section 3, the basic types of swarm behavior are modeled and operationally studied. Section 4 covers the design of the proposed swarm algorithm. In Section 5 the experimental setup is detailed, with different combinations of techniques reviewed and tested on both synthetic and real-world scenarios. Section 6 draws conclusions and future work.

2. Related Work

In this section, we present a review of the research addressing target search with swarm of UAVs. According to [16], target search is the discovery of targets located in an unstructured environment, with no prior knowledge about their location and about the obstacles layout. The quality of the process can be measured by minimizing the overall time needed for completing the mission. A target search mission with swarms of robots (agents) can be defined according to (i) the number of targets and agents; (ii) the mobility of targets; (iii) the complexity of the environment; (iv) the prior knowledge about the target; (v) the type of swarm coordination.

Several research has been developed in the field, considering various problem setups when addressing the problem of target search. Such works either describe working systems, or focus on a single specific challenge to be studied. Thus, the systems available on the literature present different degrees of maturity. Some systems have been deployed and tested in real-life scenarios, while others remain theoretical approaches. Among the proposed systems some approaches are tailored to suit requirements of a specific kind of event and are therefore domain specific. Thus a comparative performance evaluation is often not feasible due to the heterogeneity of goals, methodologies and applications. A recent survey [17] attempted to discuss the qualitative differences among the approaches using a taxonomy. For the sake of brevity, in this section some relevant works in the field are briefly summarized, discussing differences and similarities with our approach.

Table 1 shows a structured view of the techniques for coordinating swarms of UAVs according to [18]. Specifically, the *formation control* focuses on the spatial arrangement of the UAVs, and it can be distinguished as *Virtual Structure*, *Leader Following* or *Flocking*. *Virtual Structure* represents the formation by means of the reference points of a basic geometric structure [19]. It is useful when constraints on the mutual distance between UAVs should be managed [17]. However, when obstacles are present, complex and multiple maneuvers are needed to recover the predefined arrangement. With *Leader Following* the drones keep a relative distance and angle from a leader, which has a higher awareness of the search field or a better equipment [20]. However, the leader can easily become a point of failure of the entire swarm due to its control [21]. A more flexible schema is *Flocking*, which is still based on mutual distances among drones, but with a dynamic rearrangement capability based on: (i) maintaining the heading of flockmates, (ii) avoiding collisions and (iii) attempting to remain in range between them [11], [22], [12], and [23].

A *Trajectory Following* strategy aims at determining the path toward the targets by defining the swarm collective motion explicitly [24], or by using macroscopic rules such as the *artificial potential* [25],[26]. However, it requires the knowledge of the scenario layout in order to steer the swarm. Thus, it is not suitable for our research purpose. Similarly, with approaches based on *Swarm Aggregation* the ultimate goal of aggregation is to form a single aggregate that contains all the agents in a specific known location [27], [28].

Social Foraging is an example of collective behavior based on *Stigmergy*: each agent of the swarm releases a pheromone mark in a shared medium while performing a given action. The pheromone acts as a stimulus, i.e. other agents that perceive it will follow the pheromone trail toward the area where the target is located [29], [30], [31].

In the *Distributed Agreement or Consensus*, the agreement in the swarm is achieved if the variables of interest of all drones converge to the same value. These variables can represent the state of the agent, e.g. its heading or its current behavior [32], or the state of the overall scenario, e.g. the nature of the targets [33]. A consensus-based approach for collaborative UAVs search provides an evident advantage when dealing with uncertainty about the target detection process [34]. However, this situation is out of the scope of this paper.

The *Swarm Tracking* problem is based on the generalized pursuit-evasion problem for multi-agent systems. It assumes the existence of a moving target which is trying to evade the capture by the agents [35], [36], [37]. This research does not fit our requirements, since we assume that targets are static.

Table 1: Characterization and qualitative comparison of the approaches aimed at coordinating a swarm of UAVs according to [38], [16], and [39].

APPROACH	**MAIN ADVANTAGES**	**MAIN DISADVANTAGES**
Formation Control: Virtual Structure [19] [17]	Consistent swarm performance in clear search space	Complex management of obstacles. Limited flexibility and scalability
Formation Control: Leader Following [20] [21]	Flexible formations. No apriori knowledge of the search field layout	Single point of failure (the leader). No feedback from the followers to the leader to adjust the formation
Formation Control: Flocking [22] [23] [12]	Dynamic formations rearrangement. No apriori knowledge of the search field layout	Poor predictability of the overall swarm behavior
Trajectory Following and Artificial Potential [25] [26]	Efficient building of the search path	Easy to fall into local minima. Path or potential fields are explicitly defined according to search field layout
Swarm Aggregation [27] [28]	Predictable swarm performance	Requires the apriori knowledge of the targets positions. May require the synchronization among agents
Social foraging: Stigmergy [30] [29] [31]	No apriori knowledge of the search field layout	Poor predictability of the overall swarm behavior
Distributed Consensus [32] [34]	Handle uncertainty in target detection	High communication rate
Swarm Tracking [37] [35]	Handle mobile target	Complex scalability, especially with multiple targets

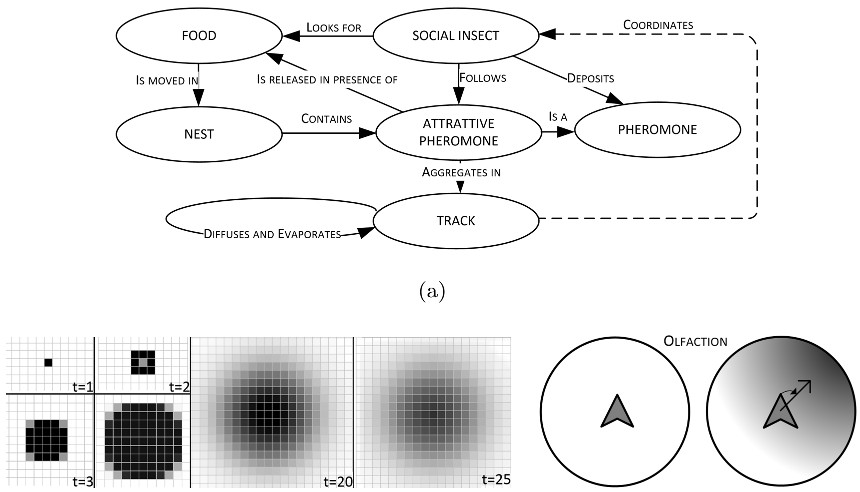
In this paper we propose a combined technique of flocking and stigmergy for the target search problem. To generate a swarm that is adaptive with respect to the scenario, an evolutionary algorithm is used for parameters tuning, [40], [41]. The tuning is optimized to minimize the time for completing the mission, producing a coordination schema that is emergent, given the use of a metaheuristic and the local interactions between agents [5], [12]. As an example in [42] authors use the information sensed from the environment to feed an evolutionary algorithm aimed at guiding the UAVs through the environment. The proposed model is called FSE, which stands for Flocking, Stigmergy, and Evolution. Among the different sub-classes of evolutionary algorithms, we consider DE, which has been used in the swarm robotics domain [13], as well as in the coordination of multiple drones. For example, in [43] drones are coordinated to navigate from a known initial position to a predetermined target location .

3. Basic types of swarm behavior

In this section the proposed FSE model is specified through the fundamental types of biologically-inspired processes adopted, namely, Stigmergy, Flocking, and Evolution. We adopt the ontology as a specification formalism, to guarantee consistency, overcome barriers to the acquisition of specifications, and promote the knowledge reuse. An ontology is a structured definition of basic concepts and relations among them [44]. The first swarm behavior is marker based stigmergy [30]. With stigmergy, biological agents leave piece of information in the environment in the form of pheromones, i.e., volatile substances locally diffused. A pheromone stays temporarily around the place where it was released. Other agents, when sensing the pheromone, can properly react and modify their behavior [45]. As an example, ant seeking for food are attracted by pheromones released by other ants, which eventually corresponds to the shortest path between the food and the nest.

Fig. 1a shows the ontology of the attractive pheromone exhibited by social insects in foraging. In ontology diagrams, base concepts are enclosed in white ovals and connected by properties, represented by black directed edges. The fundamental properties are: *Attractive Pheromone* is released in presence of *Food*, *Attractive Pheromone* aggregates in *Track*, and *Social Insect* follows *Attractive Pheromone*. The property *Track* coordinates *Social Insect* is represented by a dashed arrow because it is not directly implemented, since it is an emergent property of the system. Fig. 1b illustrates the dynamics of pheromone release: after initial release (t=1) it is first diffused (t=2, 3, etc.) and then evaporated (t=20 and 25). Fig. 1c shows the pheromone sensing and acting dynamics: here, the *social insect* called *agent* for concept abstraction follows the direction of major intensity.

A given location may be repeatedly visited multiple times by the same or by different agents, and this may lead to an ineffective search. To solve this

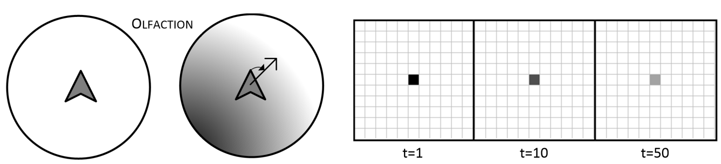


(a)

(b) (c)

Figure 1: (a) Ontology of the attractive pheromone exhibited by ants in foraging process. (b) Illustration of the pheromone release dynamics. (c) Illustration of the pheromone sensing and acting dynamics.

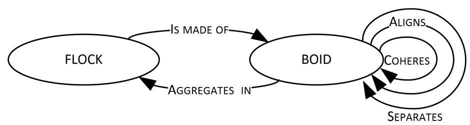
problem, the repulsive pheromone has been introduced. In essence, an agent marks target-free positions with a repulsive pheromone. The mechanism of repulsive pheromone is complementary with respect to the attractive one [10].



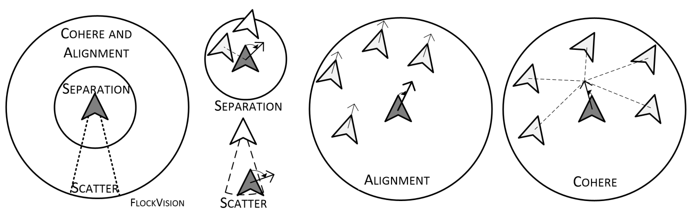
(a) (b)

Figure 2: Repulsive Pheromones Role (a) and Dynamics (b)

As represented in Fig. 2a, the agent follows the minimum intensity of the repulsive pheromone. To achieve the intended result without compromising collateral exploration, repulsive pheromones diffusion is not allowed, while evaporation behaves as usual (Fig. 2b). An attractive Pheromone peak may keep a disproportionate number of agents in a small area, which is not efficient for exploration. For this reason, olfactory habituation has been introduced [9]. More specifically, when exposed to a maximum of intensity, the pheromones sensing quickly saturates, leading to inability to sense for a while. This allows the agent to leave the saturated area.



(a)



(b)

Figure 3: (a) Flocking agents ontology. (b) Illustration of separation, alignment, cohere.

Fig. 3a shows the ontology of flocking agents, called *boids* in the literature, which is based on three features: *separation*, *alignment*, and *cohere* [11]. Each agent acts repeating the following prioritized rules. Separation: steer to avoid crowding flockmates; alignment: steer towards the heading of flockmates; cohesion: steer to move toward the center of flockmates. Flockmates are determined according to different radiuses for each rule, as represented in Fig. 3b. We have verified that: (a) the cohere is not suitable for drones moving between many obstacles; (b) the separation can be better exploited with an area different than circular, as represented in Fig. 4.

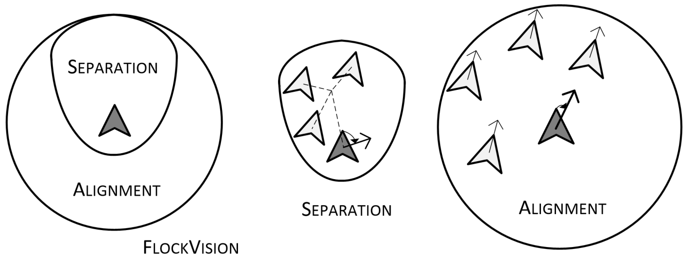
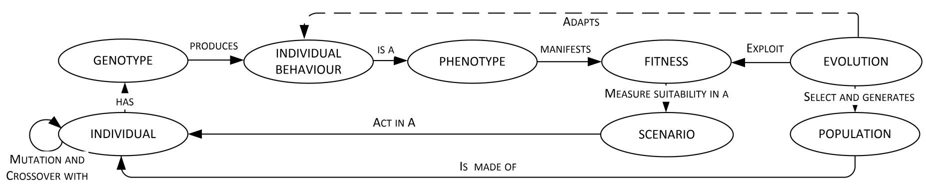
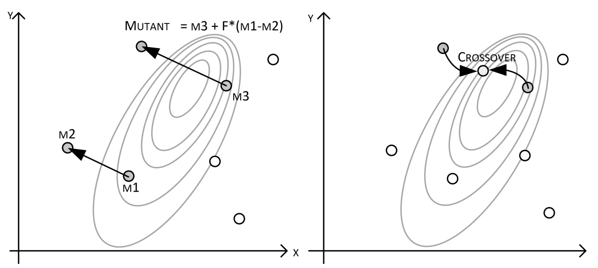


Figure 4: Flocking agent procedures for drones: search area, separate and alignment.

Fig. 5a shows the *evolution* ontology. It exploits mechanisms inspired by biological *evolution*, such as mutation, crossover, and selection. Briefly, candidate solutions to the parameterization problem play the role of *individuals* in a *population*. Each individual mutates and mates with its peer generating a new one, characterized by its own *genotype*. Individuals differently fit with the scenario. Such *fitness* is the basis for the objective function.



(a)



(b)

Figure 5: (a) Evolution Ontology. (b) Illustration of mutation and crossover.

The evolution of the population takes place after the repeated application of the above operators in order to obtain a better solution for each generation.

Among all the different approaches, we refer to DE [46] due to its simplicity and efficiency in global optimization over continuous spaces. The interested reader is referred to [13] for further details. Fig. 5b shows a simplified search space for the parameters vector. DE mutation takes the difference of two agents (multiplied by a factor F) to explore the objective function landscape, whereas DE crossover generates an offspring by mixing fragments of the current and the mutated elements.

4. Design

The basic types of swarm behavior have been combined in the FSE swarm coordination schema. For the sake of clarity, in Fig. 6 we propose a view of the SFE schema at the concept level by using an ontology, highlighting how and in which aspects the different basic types of biologically-inspired processes are combined. Specifically, *Flocks* are group of *Drones*, detecting *Targets*, and acting like *Boids*. A *Drone* consumes *Fly Time*, which is measured as *Fitness*. A *Drone* acts in a cyberphysical environment made of both *Digital* and *Physical environments*. *Attractive* (or *Repulsive*) *Pheromone* is released in presence (or absence) of a *Target*. *Attractive Pheromone* aggregates in *Track*, which is diffused and evaporated by the *Digital Environment*. *Drone* avoids *Obstacles* and *Boids* via *Steering Behavior*. Like a *social insect*, *Drone* turns away from repulsive pheromone and is attracted by attractive pheromone. The latter leads to *Olfactory Habituation*. Finally, *Evolution* selects and generates a *Population* of *Swarms*. Each *Swarm* has a *Genotype*, with a *Fitness* with respect to a *Scenario*. The *Evolution* exploits the Fitness adapting the Swarm Behavior to the environment.

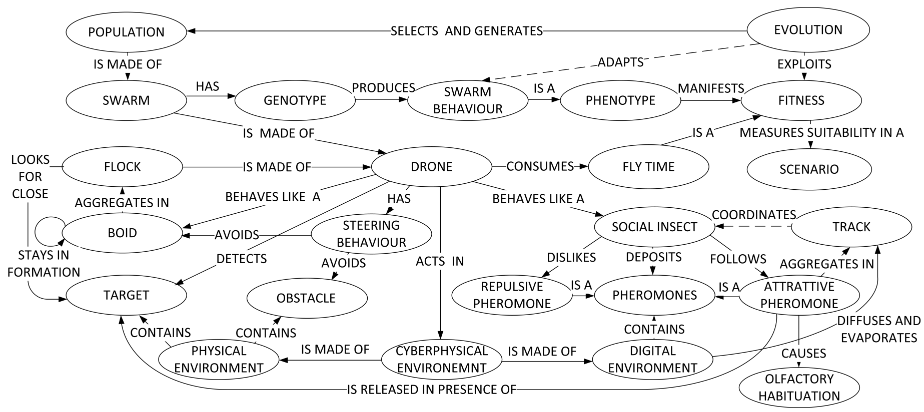


Figure 6: Ontology of the FSE swarming algorithm.

For the sake of simplicity, the digital environment is designed on a two dimensional space. Indeed, it could be inefficient and unreliable to have more than one drone on the same location. In the worst case, drones could interfere with each other degrading the performance. More operatively, the digital environment manages the aggregation, diffusion, and evaporation of the pheromones map. It can be seen as lattice of cells. When a single pheromone (intensity I) is released: (i) it is added to the underlying map; (ii) it is progressively diffused to the nearby cells, with a constant diffusion rate $\delta \in [0,1]$; (iii) it is subsequently evaporated, decreasing its intensity over time by the constant rate $\varepsilon \in [0,1]$. More formally, the pheromone intensity is the output of a scalar function $\mathrm{p}_{(x,y)}(t)$, released at the instant t on the cell (x, y), and characterized by the dynamics:

$$\mathrm{p}_{(x,y)}(t) = \varepsilon \cdot \Big[(1-\delta) \cdot \mathrm{p}_{(x,y)}(t-1) + \partial \mathrm{p}_{(x,y)}(t-1,t) + \partial \mathrm{d}_{(x,y)}(t-1,t)\Big] \quad (1)$$

where $(1-\delta) \cdot \mathrm{p}_{(x,y)}(t-1)$ is the pheromone amount remaining on the cell (x, y) after diffusion to nearby cells, whereas $\partial \mathrm{p}_{(x,y)}(t_1, t_2)$ is the additional pheromone released on the cell (x, y) in the interval (t_1, t_2), and $\partial \mathrm{d}_{(x,y)}(t_1, t_2)$ is the additional pheromone diffused from all the nearby cells to the cell (x, y)

in the interval (t_1, t_2). The total amount is also multiplied by ϵ to take into account evaporation. The diffused pheromone can be formally calculated in the interval $(t-1, t)$ as:

$$\partial \mathrm{d}_{(x,y)}(t-1,t) = \frac{\delta}{8} \cdot \sum_{i=-1}^{1} \sum_{j=-1}^{1} \mathrm{p}_{(x+i,y+j)}(t-1) \tag{2}$$

since each of the 8 neighbor cells propagates the portion δ of its pheromone to the cell (x, y) at each update cycle.

While the environment manages its dynamics, each drone behaves according to local circumstances. Table 2 summarizes the set of parameters of the FSE algorithm, which represents the swarm genotype adapted by DE. Specifically, let us consider the parameters that have non been aforementioned. In the FSE algorithm, each drone periodically carries out a target check on its location. If there is an unknown target, it releases an amount *stigIntensity* of attractive pheromone on its location. If no target is found, the drone releases repulsive pheromones. Subsequently, if there are drones or objects within *obstacleVision* radius, then the drone turns away. If there are attractive pheromones within *olfaction* radius, the drone turns toward the maximum amount of it. If drone is located at the maximum, it is subject to olfactory habituation. When there are no pheromones, if the drone detects flockmates in a separation area, it slows down and turns away from them (at most by *maxSeparateTurn*).

The separation area is created by union of a circular area of radius *minimumSeparation* around the drone, with a conical area in front of the drone, sized by *minFlockAngle* and *flockVision*. If there are no flockmates found in this area, the drone steers towards the drones in a larger circular area (with radius *flockVision*) by a quantity within *maxAlignTurn*. Subsequently, if neither drones, nor pheromones, nor obstacles are found, the drone steers towards the minimum of repulsive pheromone, if available, otherwise randomly. Finally, it moves forward. More formally, the part of the FS algorithm can be summarized by the pseudocode presented in Algorithm 1.

Table 2: Description of the parameters of the SFE algorithm

Parameter	Description (unit measure)
droneVel	Drone horizontal speed (m/s)
wiggleVar	Drone max rand-fly turn angle (°)
obstacleVision	Drone object sensing distance (m)
flockVision	Flock visibility cone radius (m)
minimumSeparation	Flock separation distance (m)
maxSeparateTurn	Flock separation turn limit (°)
minFlockAngle	Flock separation front cone (°)
maxAlignTurn	Flock alignment turn limit (°)
stigIntensity	Pheromone release intensity
stigDiffusion	Pheromone diffusion rate on surrounding cells (%)
stigEvaporation	Pheromone evaporation rate (%)
olfaction	Pheromone sensing distance (m)

Specifically, the executeFS($\cdot$) function carries out a simulated scenario. In such scenario, the *Swarm* is a set of drones D_i exploring the environment where *Obstacles* and *Targets* are statically specified. Furthermore, *AttractivePheromones* (or *RepulsivePheromones*) is a dynamic structure released when a drone is (is not) positioned on an unknown target. The drone steering is ruled, in ascending priority order, by (i) obstacles avoidance, (ii) pheromone potential, (iii) flocking formation. The set of targets already found is tracked at each instant of time (tick). Finally, executeFS($\cdot$) returns the first (i.e., the minimum) instant of time for which the percentage of targets found is greater or equal to 95%.

In order to adapt the swarm behavior, the search time returned by the FS procedure is used to feed the DE as illustrated in Algorithm 2. Let K be the number of adaptive parameters of the FSE algorithm. In DE a solution is represented by a real K-dimensional vector. As summarized in Table 2, the set of parameters of the FSE algorithm includes various typologies: actuation, sensing,

Algorithm 1: Coordination algorithm based on Flocking and Stigmergy

```
function executeFS(Swarm, Obstacles,
  Targets,AttractivePheromones,RepulsivePheromones)
tick = 0; cardT_F = 0; cardT = count(Targets) ;
do
    for each drone D_i in Swarm do
        if (D_i.position in Targets) then
            markTargetAsFound(Targets, D_i.position); cardT_F = cardT_F + 1;
            releasePheromone(AttractivePheromones, D_i.position);
        else
            releasePheromone(RepulsivePheromones, D_i.position);
        if (D_i.obstacleVision() intersects Obstacles) then
            turnDrone(D_i.heading);
            else if (D_i.olfaction() intersects AttractivePheromones) then
                turnDroneTowardsMax(D_i.heading,AttractivePheromones);
                else if (D_i.flockmates in D_i.separateArea()) then
                    turnDroneWithin(D_i.heading, maxSeparateTurn);
                    else if (D_i.heading != D_i.flockmates().meanHeading)
                     then
                        turnDroneWithin(D_i.heading,maxAlignTurn);
                        else if (D_i.olfaction() intersects
                         RepulsivePheromones) then
                            turnDroneTowardsMin(D_i.heading,RepulsivePheromones);

                        else turnDroneWithin(D_i.heading, wiggleWar);
        moveForward(D_i.pos,D_i.heading,D_i.vel);
    evaporatePheromone(AttractivePheromones,RepulsivePheromones);
    tick = tick + 1;
while cardT_F < 0.95 · cardT;
return tick;
```

flocking and stigmergy. DE starts with a randomly generated population of N candidate solutions, i.e. $P^{(0)}$. At each iteration t and for each genotype $\boldsymbol{p}_i^{(t)}$ of the current population $P^{(t)}$, a mutant vector $\boldsymbol{m}$ is created by applying the mutation of randomly selected members. Then, a trial vector $\boldsymbol{p}_i^{*}$ is created by crossover of $\boldsymbol{m}$ and $\boldsymbol{p}_i^{(t)}$. Subsequently, the population is modified according to the best fitting vector between the fitness of the trial vector (f_i^*) and the fitness of the initial genotype ($f_i^{(t)}$). When the termination criterion is true, the vector characterizing the solution with the best fitness (i.e. the shortest search time) in the current population is considered as the optimal swarm parametrization.

More formally the fitness can be defined as following. Given a simulated scenario Σ, made of: (i) simulation instants of time $t\ \epsilon \mathbb{N}^{+}$; (ii) a set of drones $\{D_i\}$, each drone having a dynamic position $(x_t, y_t)_{Di}$; (iii) a set of targets $\tau \epsilon \mathrm{T}$, each target having a fixed position $(x, y)_\tau$. The set of targets already found $\mathrm{T}_F(t) \subseteq \mathrm{T}$, at a given instant of time t, is the set of targets for which it exists a time $t' \leq t$ and a drone d such that the drones position corresponds to the targets position:

$$\mathrm{T}_F(t) = \{\tau | \exists D_i, \exists t' \leq t : (x_{t'}, y_{t'})_{Di} = (x, y)_\tau\} \tag{3}$$

The fitness of the simulated scenario Σ is the minimum instant of time for which $\mathrm{T}_F(t)$ has cardinality greater than or equal to 0.95 of the cardinality of T:

$$\text{fitness}(\Sigma) = min_{t\epsilon\mathbb{N}^+}\{t : |\mathrm{T}_F(t)| \geq 0.95 \cdot |\mathrm{T}|\} \tag{4}$$

Algorithm 2: Differential Evolution Algorithm

function DifferentialEvolution($Obstacles$, $Targets$, $AttractivePheromones$, $RepulsivePheromones$)

$t = 0$;

$P^{(0)} =$ initializePopulation();

for each genotype $\boldsymbol{p}_i^{(0)}$ **in** $P^{(0)}$ **do**

  $S_i^{(0)} =$ genotypeToSwarm($\boldsymbol{p}_i^{(0)}$);

  $f_i^{(0)} =$ executeFS($S_i^{(0)}$, $Obstacles$, $Targets$, $AttractivePheromones$, $RepulsivePheromones$);

do

  for each genotype $\boldsymbol{p}_i^{(t)}$ **in** $P^{(t)}$ **do**

    $\boldsymbol{m} =$ generateMutant($P^{(t)}$, $\boldsymbol{p}_i^{(t)}$);

    $\boldsymbol{p}_i^* =$ binomialCrossover($\boldsymbol{p}_i^{(t)}$, $\boldsymbol{m}$);

    $S_i^* =$ genotypeToSwarm($\boldsymbol{p}_i^*$);

    $f_i^* =$ executeFS(S_i^*, $Obstacles$, $Targets$, $AttractivePheromones$, $RepulsivePheromones$);

  for each genotype $\boldsymbol{p}_i^{(t)}$ **in** $P^{(t)}$ **do**

    if $(f_i^* < f_i^{(t)})$ **then**

      $\boldsymbol{p}_i^{(t+1)} = \boldsymbol{p}_i^*$; $f_i^{(t+1)} = f_i^*$;

    else

      $\boldsymbol{p}_i^{(t+1)} = \boldsymbol{p}_i^{(t)}$; $f_i^{(t+1)} = f_i^{(t)}$;

    $f_{min}^{(t+1)} = min\{f_1^{(t+1)}, ..., f_N^{(t+1)}\}$;

  $t = t + 1$;

while (terminationCriterion($f_{min}^{(t)}$,t) = **false**);

return genotypeToSwarm($\boldsymbol{p}_{min}^{(t)}$);

DE itself has at least 2 hyperparameters: the scaling factor $F \epsilon [0, 2]$ from which results the mutant vector, and the crossover probability CR. The smaller CR the higher probability to produce a vector that is more similar to the target vector rather than to the mutant vector. More formally, Algorithm 3 and

Algorithm 4 define the mutation and the crossover operators, respectively:

Algorithm 3: Mutant vector generation

function generateMutant($P^{(t)}$, $\boldsymbol{p}_i^{(t)}$)

$\boldsymbol{p}'$ = randomExtraction($P^{(t)} \setminus \{\boldsymbol{p}_i^{(t)}\}$);

$\boldsymbol{p}''$ = randomExtraction($P^{(t)} \setminus \{\boldsymbol{p}_i^{(t)}, \boldsymbol{p}'\}$);

$\boldsymbol{p}'''$ = randomExtraction($P^{(t)} \setminus \{\boldsymbol{p}_i^{(t)}, \boldsymbol{p}', \boldsymbol{p}''\}$);

return $\boldsymbol{p}' + F\cdot$ ($\boldsymbol{p}''$ - $\boldsymbol{p}'''$);

Algorithm 4: Binomial crossover

function binomialCrossover($\boldsymbol{p}_i^{(t)}$, $\boldsymbol{m}$)

k = randomInteger(1, K);

for each j-th gene $p_{j,i}^{(t)}$ **in** $\boldsymbol{p}_i^{(t)}$ **do**

  if (randomReal(0,1) $< CR$) or ($(j = k)$) **then**

    $w_j = m_j$;

  else

    $w_j = p_{j,i}^{(t)}$;

return $\boldsymbol{w}$;

To show the efficiency of the proposed approach, it follows a time complexity analysis of Algorithm 1 and Algorithm 2. Let D be the number of drones, C the number of cells of the environment, and T the number of ticks of a simulation. The *for* instruction in Algorithm 1 contains a constant number of simple instructions and the *releasePheromone* instruction, which can involve at most C simple instructions when releasing the largest pheromone. That *for* instruction is executed D times, and then the asymptotic time complexity is *O(DC)*. The *repeat* instruction is executed T times. It includes the *for* instruction and the *evaporatePheromone* instruction, which can involve exactly C simple instructions when the pheromone is disperse over the entire area. Thus, the asymptotic time complexity of Algorithm 1 is *O(T(DC+C))* = *O(TDC)*.

Let P the population size and G the number of generations. In Algorithm 2, the *for* instruction is executed P times, whereas the repeat instruction G times. Thus, the asymptotic time complexity of Algorithm 2 is *O(GPTDC)*. Typical values are $G = 20$, $P = 20$; $T = 1000$, $D = 80$, and $C = 400$. To give an order of magnitude of a single execution of Algorithm 1, table 3 shows the values in seconds, calculated over 10 runs for each scenario. Here, the following hardware/software was used: CPU Intel Core i7-6700HQ at 2.60-3.50 GHz, 6M Cache, 16 GB DDR3L 1600MHz RAM, Windows 10 OS.

Table 3: Execution time of Algorithm 1, over 10 runs, for each scenario.

Dump	Field	Forest	Rural Mine	Urban	Urban Mine
$50 \pm 8sec$	$35 \pm 5sec$	$42 \pm 8sec$	$44 \pm 6sec$	$49 \pm 2sec$	$54 \pm 1sec$

Thus, considering the average value of 50 sec. for Algorithm 1, the order of magnitude of Algorithm 2 is $G \cdot P \cdot 50sec = 20 \cdot 20 \cdot 50sec = 20000sec = 5.5hours$ for the parameterization of each scenario.

The next section focuses on experimental setup and results.

5. Experimental Setup and Results

The FSE algorithm has been implemented using NetLogo, a leading simulation platform for swarm intelligence [47] and Matlab, an algorithmic development framework, for the adaptation algorithm[48], respectively. It is assumed that each drone is equipped with wireless communication capability, self-location capability, one or more target sensing technology, processor with limited computing capability, and obstacle avoidance capability. As a reference area size, let us consider 4000 square meters. To make realistic assumptions, some requirements and constraints of sensors and drones have been studied, such as the time limit and drone velocity. In [49] gas detection is provided via drones using a 1Hz rate sensor. In [50] drones to detect mines using image recognition through camera allow to have 80% recognition precision flying at 2.2 m/s. To fit among different needs of each scenario and to raise the precision of our recognition, the drone velocity is set to 1 m/s. Assuming a drone battery life of 25 minutes, the fly time is actually lower, due to the time needed to return to the place where drones are gathered [8]. Regarding the evolutionary approach, in this work, we refer to [13] and use the most common DE variant, called "DE/rand/1/bin" according to a naming convention known as "DE/x/y/z". In this convention, "DE" stands for differential evolution, "x" the base vector to be perturbed, "y" is the number of difference vectors considered for perturbation of "x", and "z" the type of crossover being used. Thus, the used variant is characterized by a perturbation with randomly ("rand") selected members, only one ("1") weighted difference vector, and a binomial ("bin") crossover used in conjunction. Apart from the DE variant, some meta-parameters must also be chosen. Specifically, in the optimization process, the differential weight F belongs to [0, 2], and mediates the generation of the mutant vector. The component of a child vector is taken with probability CR from the mutant vector and with probability 1-CR from the target vector. In [31] an early application of DE is studied and proposed to parameterize the coordination of a group of drones whose schema is obtained by employing flocking and digital pheromones. The provided results reveal as good value for CR and F are respectively 0.5 and 0.7. Fig. 7 shows the six scenarios that have been considered. Each of them is included in a squared area of 200 meters side length. More specifically:

(a) The Illegal Dump scenario is based on the Abusive Trash Map in Paterno (Sicily), and is composed by 140 trees, 19 differently-sized buildings, 11 groups of targets with an average of 4 targets per group [51].

(b) The Field scenario is a synthetic scenario made by 5 clusters of targets scattered over the area, with about 10 targets per group. There are no obstacles, 80 total drones arranged into 4 swarms, represented by triangular forms, and are placed at the antipodes of the area. Cluster of dots represent targets. There are neither obstacles nor buildings.

(c) The Forest scenario represents a synthetic reconstruction of spread targets in a stand of timber. Here, 20 targets, 400 trees (single obstacles), 80 total drones, arranged into 4 swarms, have been initially placed at the antipodes of the area.

(d) Urban Mine scenarios is derived from real-world examples of areas near

Sarajevo, in Bosnia-Herzegovina, with landmine objects, selected from publicly available data [52]. It is made by 40 targets, 59 trees and 28 buildings.

(e) The Urban scenario is characterized by two clusters of 110 total targets placed on two sides of 7 total buildings (represented by an area filled with single obstacles side by side). 40 drones, arranged into 4 swarms, are placed at the antipodes of the area, with no trees at all.

(f) The Rural Mine is derived from real-world examples of areas near Sarajevo, as well as Urban Mine [52]. It is composed by 28 target, 281 trees and 3 buildings.

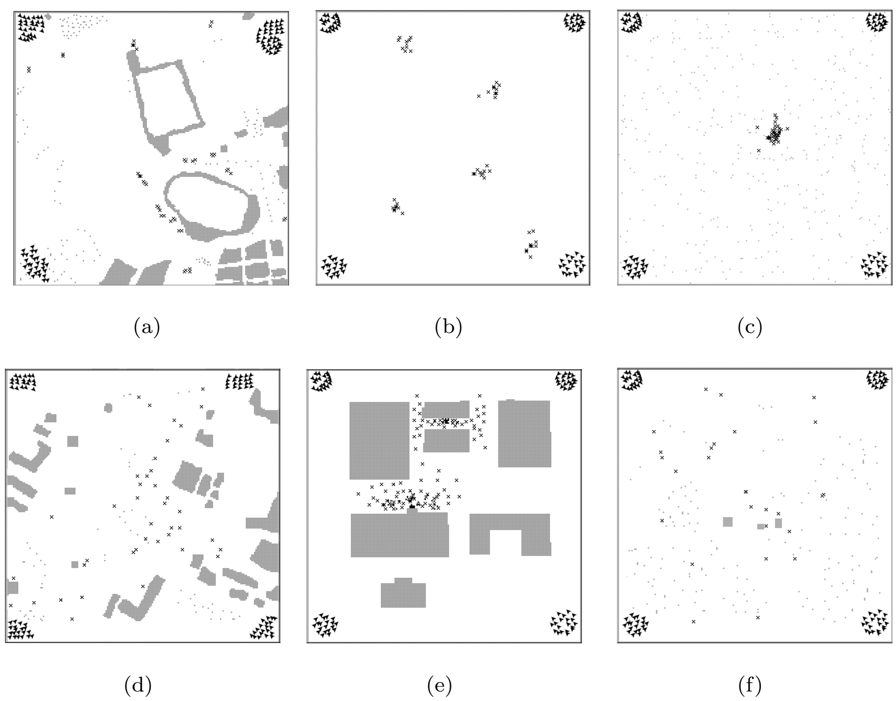


Figure 7: Scenarios Layout: (a) Illegal Dumps, (b) Field, (c) Forest, (d) Urban Mine, (e) Urban, (f) Rural Mine.

To obtain the best performance algorithm, we have experimented a number of variants of the proposed approach. More specifically: (i) in the $F_{ASC}S_AE$ algorithm, Flocking includes align, separation and cohere, Stigmergy includes attractive pheromone and finally Evolution is also available; (ii) in the $F_{AS}S_AE$ algorithm the cohere [11] has been removed from Flocking; (iii) In the $F_{ASS}S_AE$ the scatter procedure (Fig.3b) has been incorporated into the separate flocking procedure (Fig.4) [31]; (iv) in the $F_{ASS}S_{AR}E$ the repulsive pheromone has been also added [12]; (v) in the $F_{ASS}S_{ARO}E$, to further speed the search, the olfactory habituation procedure has been added. Table 4 shows the performance of the different variants, in terms of 95% confidence interval over 10 trials of completion time (ticks) on each scenario.

Table 4: Performance of different variants of the FSE algorithm.

Version	Dump	Field	Forest	Rural Mine	Urban	Urban Mine
$F_{ASC}S_AE$	1043 ± 133	791 ± 109	569 ± 157	1658 ± 247	933 ± 218	1595 ± 296
$F_{AS}S_AE$	895 ± 179	507 ± 58	425 ± 36	1693 ± 276	818 ± 176	1356 ± 191
$F_{ASS}S_AE$	866 ± 205	546 ± 101	416 ± 76	1030 ± 123	932 ± 122	880 ± 59
$F_{ASS}S_{AR}E$	743 ± 45	511 ± 37	400 ± 28	981 ± 94	720 ± 56	878 ± 53
$F_{ASS}S_{ARO}E$	638 ± 33	356 ± 47	331 ± 25	783 ± 84	562 ± 33	846 ± 74

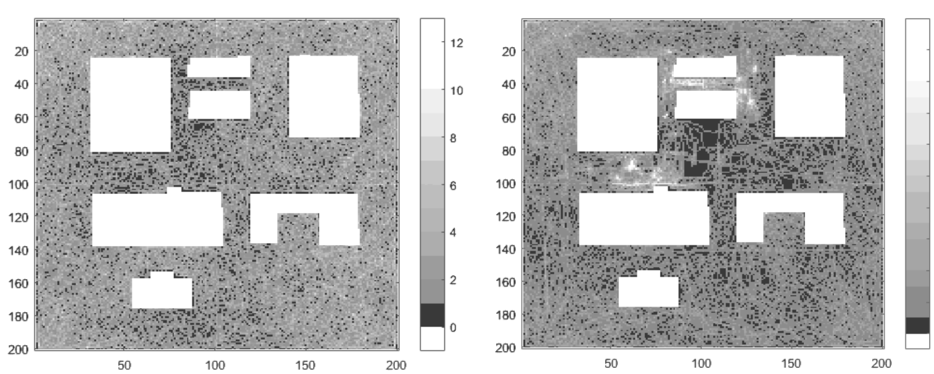
Results in Table 4 clearly shows that the different variants give an incremental improvement of performance. Considering the average performance over all scenarios, the improvement from the first to the last variant is 47% in all scenarios.

Table 5 shows the adaptive capability of the $F_{ASS}S_{ARO}E$ algorithm. Here, the Maximum Pheromone Diffusion represents the diameter of the area covered by a single pheromone deposit at its maximum diffusion, i.e., the pheromone size. Minimum Separation is the minimum distance between flockmates. Finally, Flock Vision is the maximum distance between flockmates. It is worth noting which strategy has been implemented by the DE. More specifically, in Forest and Urban scenarios the pheromone is larger than the distance between flockmates, so as to attract extra-flock drones. Indeed, both scenarios contain few and dense clusters of targets. Here, to attract additional drones with respect to flock mates is a good strategy. In contrast, the other scenarios contain targets scattered over the area. Here, to attract only flockmates is a good strategy.

Table 5: Maximum pheromone diffusion against flock vision after adaptation phase

unit(cells)	Dump	Field	Forest	Rural Mine	Urban	Urban Mine
Max Pherom. Diffusion	7	6	9	3	9	5
Minimum Separation	2.8	2.9	1.9	1.9	2.8	1.9
Flock Vision	7.4	7.4	7.4	3.8	7.4	5.6

Fig. 8a and Fig. 8b show a qualitative example of exploration of the Urban Scenario in 800 ticks, by using a basic random exploration and the SFE algorithm, respectively.



(a) (b)

Figure 8: Qualitative coverage analysis. (a) Targets found by a random exploration in 800 ticks: 81.3%. (b) Targets found by the FSE algorithm in 800 ticks: 94.6%

Clearly, the random exploration leads to a more inefficient exploration, visiting the most of the cells 4-8 times. On the other hand, the FSE algorithm focuses the search on the areas of interest (i.e. close to the target) which are visited 3 to 4 times more than the rest of the environment.

In general, an effective efficiency indicator is the plot of the number of cells

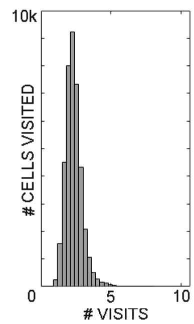


Figure 9: A coverage analysis for the Field scenario.

visited against the number of visits, averaged over 10 trials. As an example, Fig. 9 shows this indicator for the Field scenario. In general, a good exploration produces a narrower and a more-to-the-left distribution. More formally, let $v^{(i)}_{(x,y)}$ be the number of visits that the cell (x, y) receives in the i-th trial. The average over n trials is:

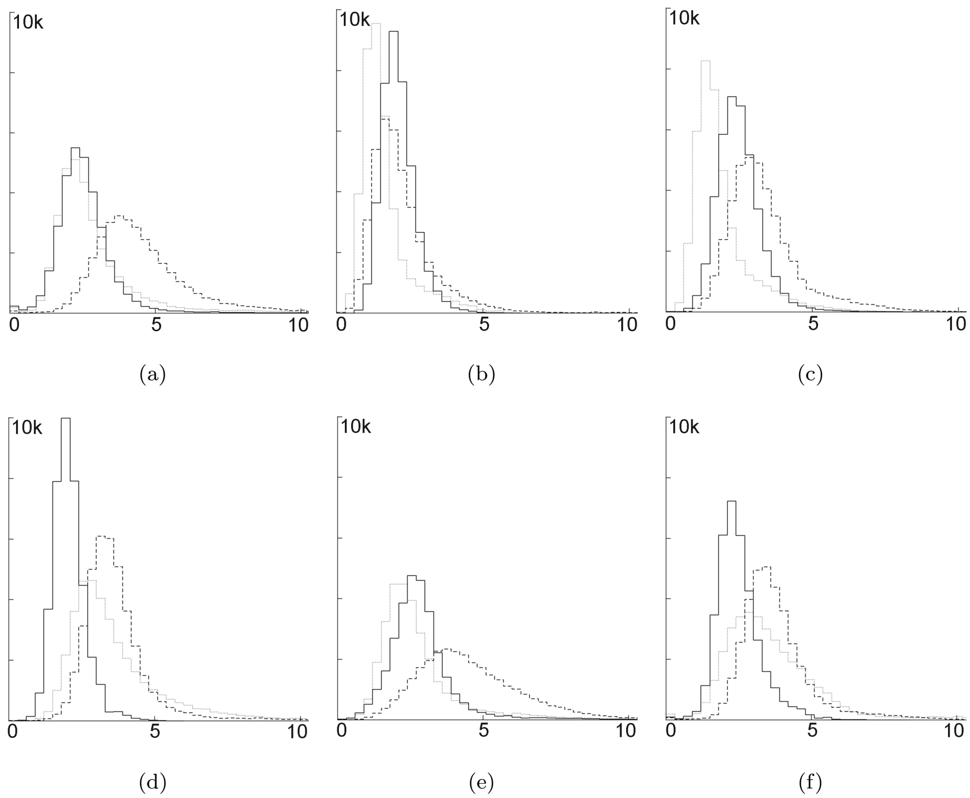
$$\lambda_{(x,y)} = \frac{1}{n} \cdot \sum_{i=0}^{n} v^{(i)}_{(x,y)} \tag{5}$$

Finally, we define the set of cells with the same average number of visits over n trials.

$$C_k = \big\{(x, y)|\lambda_{(x,y)} = k\big\} \tag{6}$$

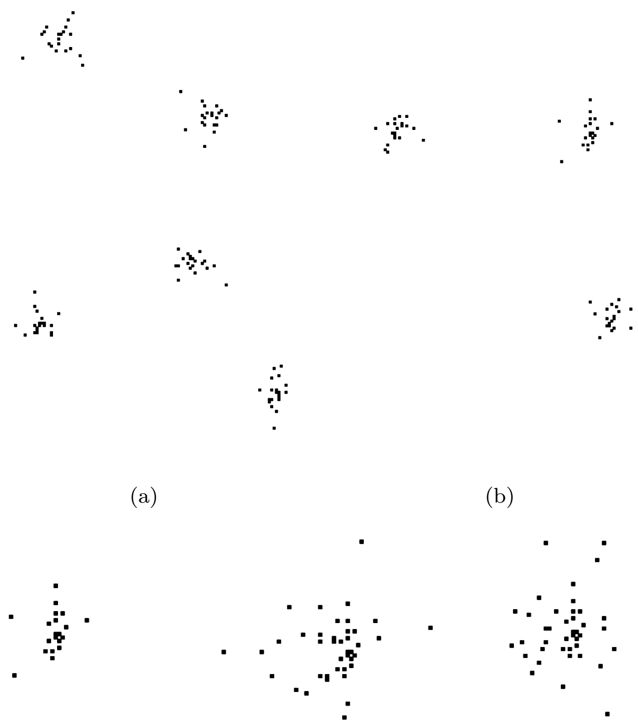
Fig. 9 is the plot of C_k as a function of k. Fig 10 shows the results for each scenario variants of FSE: $F_{ASS}S_A E$ (dashed), $F_{ASS}S_{AR}E$ (dotted), and $F_{ASS}S_{ARO}E$ (solid). It is apparent that the $F_{ASS}S_{AR}E$ approach produces the worst exploration in all cases. Indeed, the dotted curve is usually the widest and more-to-the-right distribution. In contrast, the $F_{ASS}S_{ARO}E$ algorithm produces best or close-to-the-best exploration, because the solid curve is usually a narrow and more-to-the left distribution. This clearly demonstrates the advantages of the olfactory habituation on the exploration efficiency.

Another important aspect of the algorithm is its scalability against the complexity of the scenario, in terms of distribution and number of cluster, and



(a) (b) (c) (d) (e) (f)

Figure 10: Exploration efficiency of the FSE algorithm variants on each scenario with $F_{ASS}S_{AR}E$ (dotted), $F_{ASS}S_{A}E$ (dashed), and $F_{ASS}S_{ARO}E$ (solid). (a) Illegal Dumps, (b) Field, (c) Forest, (d) Urban Mine, (e) Urban, (f) Rural Mine.



(a) (b)

(c) (d) (e)

Figure 11: Synthetic scenarios of different complexity: (a) 5 clusters, (b) 3 clusters, (c) 1 cluster, (d) 1 very big cluster, (e) 1 big cluster.

number of drones. For this purpose, Fig. 11 shows five synthetic scenarios of different complexity, with Gaussian distribution of targets.

For each scenario the search task is executed 10 times with different number of drones (20, 30, 40, 80, 120, 160¿, 200, 240), initially distributed on the scenario corners. Fig. 12 shows the average accomplishment time against drones' number. We remark that an increase of the number of drones, after about 100, does not produce significant performance improvements.

Table 6 shows the performance in terms of search time (95% confidence intervals) before and after adaptation, for each scenario. It can be observed that adaptation sensibly improves performances.

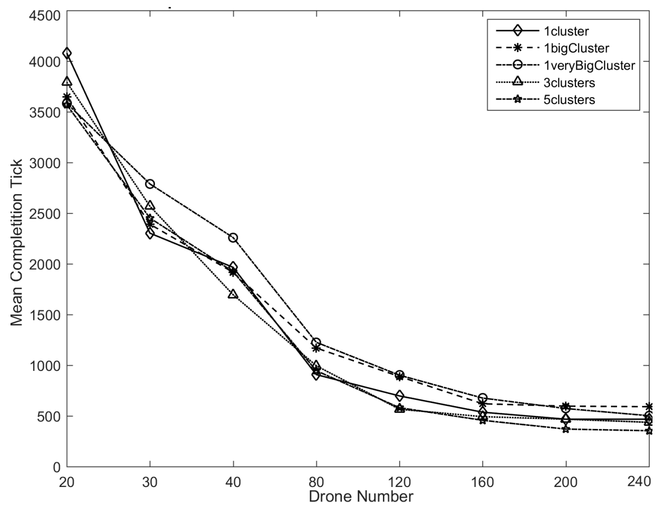


Figure 12: Average Accomplishment Time against number of drones and target distribution complexity.

Table 6: Performance before and after the Adaptation

Scenario	Before Adaptation	After Adaptation
Dumps	1132 ± 122	927 ± 71
Field	1149 ± 162	500 ± 55
Forest	1189 ± 128	515 ± 65
Rural Mine	1029 ± 59	947 ± 57
Urban	1224 ± 198	644 ± 51
Urban Mine	1228 ± 99	1109 ± 67

As a final outcome, a performance comparison with previous approaches
495 based on flocking and stigmergy is presented in Table 7, namely *Basic S+F* [12] and *Adaptive S+F** [31].

Table 7: Performance comparison with other approaches in the literature.

Scenario	*Basic S+F* [12]	*Adaptive S+F** [31]	FSE approach
Dump	934 ± 216	757 ± 112	638 ± 33
Field	589 ± 86	582 ± 121	356 ± 47
Forest	602 ± 124	593 ± 146	331 ± 25
Rural Mine	1530 ± 225	1123 ± 116	783 ± 84
Urban	890 ± 93	666 ± 100	562 ± 33
Urban Mine	1704 ± 225	1025 ± 76	846 ± 74

Considering the average performance over all scenarios, the improvement from the best performance (provided by *AdaptiveS+F**) and the proposed approach is 26%.

6. Conclusion

In this paper, we have presented a novel strategy for coordinating swarms of drones performing target search, based on three basic coordination mechanisms: stigmergy, flocking, and evolution. Each mechanism is first ontologically defined and then formally designed. Experiments have been carried out with different algorithmic variants combining various mechanisms over three synthetic and three real-world scenarios. As a result, the best variant combines: (i) flocking with aling, separation and scatter; (ii) attractive and repulsive stigmergy with olfactory habituation; (iii) differential evolution. Considering the average performance over all scenarios, the improvement from the worst to the best variant is 47%. Another result has been discovered considering the parameters set by differential evolution. Such studies have revealed that (i) performance after adaptation sensibly improves; (ii) for scenarios with few and dense targets, the best strategy is to set the pheromone size larger than the distance between flockmates so as to attract drones external to flock; (iii) for scenarios with scattered targets the best strategy is to set the pheromone size to a smaller value so as to attract only flockmates; (iv) for scenarios with multiple scattered obstacles the best strategy results in more compact flocks, to avoid flock decomposition. Further analyses on the quality of exploration have shown that the proposed algorithm clearly reduces the redundancy of the exploration. On the other hand, experiments for increasing number of drones have shown that to use more than one hundred drones does not produce significant improvements. Finally, a numerical comparison shows that the best variant proposed in this paper outperforms the other variants already available in the literature. Future work will be focused on (i) enhancing biologically inspired mechanism to allow a more efficient interoperability and adaptation; (ii) extending the sensing and actuation capabilities to allow a more realistic model of the equipment technology.

Acknowledgements

This work was carried out in the framework of the SCIADRO project, co-funded by the Tuscany Region (Italy) under the Regional Implementation Programme for Underutilized Areas Fund (PAR FAS 2007-2013) and the Research Facilitation Fund (FAR) of the Ministry of Education, University and Research (MIUR).

References

[1] K. Whitehead, C. H. Hugenholtz, S. Myshak, O. Brown, A. LeClair, A. Tamminga, T. E. Barchyn, B. Moorman, B. Eaton, Remote sensing of the environment with small unmanned aircraft systems (uass), part 2: scientific and commercial applications, Journal of unmanned vehicle systems 2 (3) (2014) 86–102.

[2] M. Lega, J. Kosmatka, C. Ferrara, F. Russo, R. Napoli, G. Persechino, Using advanced aerial platforms and infrared thermography to track environmental contamination, Environmental Forensics 13 (4) (2012) 332–338.

[3] M. Lega, C. Ferrara, G. Persechino, P. Bishop, Remote sensing in environmental police investigations: aerial platforms and an innovative application of thermography to detect several illegal activities, Environmental monitoring and assessment 186 (12) (2014) 8291–8301.

[4] R. McCune, R. Purta, M. Dobski, A. Jaworski, G. Madey, A. Madey, Y. Wei, M. B. Blake, Investigations of dddas for command and control of uav swarms with agent-based modeling, in: Simulation Conference (WSC), 2013 Winter, IEEE, 2013, pp. 1467–1478.

[5] M. Brambilla, E. Ferrante, M. Birattari, M. Dorigo, Swarm robotics: a review from the swarm engineering perspective, Swarm Intelligence 7 (1) (2013) 1–41.

[6] G. Baldassarre, V. Trianni, M. Bonani, F. Mondada, M. Dorigo, S. Nolfi, Self-organized coordinated motion in groups of physically connected robots, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 37 (1) (2007) 224–239.

[7] D. Wang, H. Wang, L. Liu, Unknown environment exploration of multi-robot system with the fordpso, Swarm and Evolutionary Computation 26 (2016) 157–174.

[8] G. Ermacora, A. Toma, B. Bona, M. Chiaberge, M. Silvagni, M. Gaspardone, R. Antonini, A cloud robotics architecture for an emergency management and monitoring service in a smart city environment, in: Intelligent Robots and Systems (IROS), International Conference on, IEEE/RSJ, 2013, pp. 1–4.

[9] D. L. Glanzman, Olfactory habituation: Fresh insights from flies, Proceedings of the National Academy of Sciences 108 (36) (2011) 14711–14712.

[10] H. Amrein, Pheromone perception and behavior in drosophila, Current opinion in neurobiology 14 (4) (2004) 435–442.

[11] C. W. Reynolds, Flocks, herds and schools: A distributed behavioral model, in: ACM SIGGRAPH computer graphics, Vol. 21, ACM, 1987, pp. 25–34.

[12] M. G. Cimino, A. Lazzeri, G. Vaglini, Combining stigmergic and flocking behaviors to coordinate swarms of drones performing target search, in: Information, Intelligence, Systems and Applications (IISA), 2015 6th International Conference on, IEEE, 2015, pp. 1–6.

[13] S. Das, P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE transactions on evolutionary computation 15 (1) (2011) 4–31.

[14] V. Kachitvichyanukul, Comparison of three evolutionary algorithms: Ga, pso, and de, Industrial Engineering and Management Systems 11 (3) (2012) 215–223.

[15] M. G. Cimino, A. Lazzeri, G. Vaglini, Improving the analysis of context-aware information via marker-based stigmergy and differential evolution, in: International Conference on Artificial Intelligence and Soft Computing, Springer, 2015, pp. 341–352.

[16] M. Senanayake, I. Senthooran, J. C. Barca, H. Chung, J. Kamruzzaman, M. Murshed, Search and tracking algorithms for swarms of robots: A survey, Robotics and Autonomous Systems 75 (2016) 422–434.

[17] Y. Zhang, H. Mehrjerdi, A survey on multiple unmanned vehicles formation control and coordination: Normal and fault situations, in: Unmanned Aircraft Systems (ICUAS), 2013 International Conference on, IEEE, 2013, pp. 1087–1096.

[18] V. Gazi, K. M. Passino, Swarm coordination and control problems, in: Swarm Stability and Optimization, Springer, 2011, pp. 15–25.

[19] L. Krick, M. Broucke, B. Francis, Getting mobile autonomous robots to form a prescribed geometric arrangement, in: Recent Advances in Learning and Control, Springer, 2008, pp. 149–159.

[20] H. Qiu, H. Duan, Pigeon interaction mode switch-based uav distributed flocking control under obstacle environments, ISA transactions 71 (2017) 93–102.

[21] A. Dang, J. Horn, Formation control of leader-following uavs to track a moving target in a dynamic environment, Journal of Automation and Control Engineering Vol 3 (1) (2015) 1–8.

[22] S. Hauert, S. Leven, M. Varga, F. Ruini, A. Cangelosi, J.-C. Zufferey, D. Floreano, Reynolds flocking in reality with fixed-wing robots: communication range vs. maximum turning rate, in: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, IEEE, 2011, pp. 5015–5020.

[23] M. De Benedetti, F. D'Urso, G. Fortino, F. Messina, G. Pappalardo, C. Santoro, A fault-tolerant self-organizing flocking approach for uav aerial survey, Journal of Network and Computer Applications 96 (2017) 14–30.

[24] A. Viguria, I. Maza, A. Ollero, Distributed service-based cooperation in aerial/ground robot teams applied to fire detection and extinguishing missions, Advanced Robotics 24 (1-2) (2010) 1–23.

[25] Y. Qu, Y. Zhang, Y. Zhang, A uav solution of regional surveillance based on pheromones and artificial potential field theory, in: Unmanned Aircraft Systems (ICUAS), 2015 International Conference on, IEEE, 2015, pp. 380–385.

[26] Q. Luo, H. Duan, An improved artificial physics approach to multiple uavs/ugvs heterogeneous coordination, Science China Technological Sciences 56 (10) (2013) 2473–2479.

[27] D. B. Wilson, A. Goktogan, S. Sukkarieh, Uav rendezvous: From concept to flight test, in: Australasian Conference on Robotics and Automation (ACRA), Vol. 25, 2012, pp. 1–10.

[28] O. Soysal, E. Şahin, A macroscopic model for self-organized aggregation in swarm robotic systems, in: International Workshop on Swarm Robotics, Springer, 2006, pp. 27–42.

[29] T. Chi, H. Cheng, J. Page, N. Ahmed, Evolving swarm of uavs, Advances in aircraft and spacecraft science 1 (2) (2014) 219–232.

[30] J. A. Sauter, R. Matthews, H. Van Dyke Parunak, S. A. Brueckner, Performance of digital pheromones for swarming vehicle control, in: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, ACM, 2005, pp. 903–910.

[31] M. G. Cimino, A. Lazzeri, G. Vaglini, Using differential evolution to improve pheromone-based coordination of swarms of drones for collaborative target detection., in: ICPRAM, 2016, pp. 605–610.

[32] H. Kurdi, J. How, G. Bautista, Bio-inspired algorithm for task allocation in multi-uav search and rescue missions, in: AIAA Guidance, Navigation, and Control Conference, 2016, p. 1377.

[33] J. Han, Y. Chen, Multiple uav formations for cooperative source seeking and contour mapping of a radiative signal field, Journal of Intelligent & Robotic Systems 74 (1-2) (2014) 323–332.

[34] R. Sharma, M. Kothari, C. N. Taylor, I. Postlethwaite, Cooperative target-capturing with inaccurate target information, in: American Control Conference (ACC), 2010, IEEE, 2010, pp. 5520–5525.

[35] G. Lee, N. Y. Chong, H. Christensen, Tracking multiple moving targets with swarms of mobile robots, Intelligent Service Robotics 3 (2) (2010) 61–72.

[36] Y. Cui, J. Ren, W. Du, J. Dai, Uav target tracking algorithm based on task allocation consensus, Journal of Systems Engineering and Electronics 27 (6) (2016) 1207–1218.

[37] S. Minaeian, J. Liu, Y.-J. Son, Vision-based target detection and localization via a team of cooperative uav and ugvs, IEEE Transactions on systems, man, and cybernetics: systems 46 (7) (2016) 1005–1016.

[38] Z. Yan, N. Jouandeau, A. A. Cherif, A survey and analysis of multi-robot coordination, International Journal of Advanced Robotic Systems 10 (12) (2013) 399.

[39] V. Gazi, K. M. Passino, Swarm coordination and control problems, in: Swarm Stability and Optimization, Springer, 2011, pp. 15–25.

[40] C. Ramirez-Atencia, M. D. R-Moreno, D. Camacho, Handling swarm of uavs based on evolutionary multi-objective optimization, Progress in Artificial Intelligence 6 (3) (2017) 263–274.

[41] R. Y. Albert, B. B. Thompson, R. J. Marks, Competitive evolution of tactical multiswarm dynamics, IEEE Transactions on Systems, Man, and Cybernetics: Systems 43 (3) (2013) 563–569.

[42] G. Varela, P. Caamaño, F. Orjales, Á. Deibe, F. López-Peña, R. J. Duro, Autonomous uav based search operations using constrained sampling evolutionary algorithms, Neurocomputing 132 (2014) 54–67.

[43] A. N. Brintaki, I. K. Nikolos, Coordinated uav path planning using differential evolution, Operational Research 5 (3) (2005) 487–502.

[44] K. Siegemund, E. J. Thomas, Y. Zhao, J. Pan, U. Assmann, Towards ontology-driven requirements engineering, in: Workshop semantic web enabled software engineering at 10th international semantic web conference (ISWC), Bonn, 2011.

[45] H. V. Parunak, M. Purcell, R. O'Connell, Digital pheromones for autonomous coordination of swarming uav's, in: 1st UAV Conference, 2002, p. 3446.

[46] K. Price, R. M. Storn, J. A. Lampinen, Differential evolution: a practical approach to global optimization, Springer Science & Business Media, 2006.

[47] Netlogo (Sep. 2017).
URL ccl.northwestern.edu/netlogo/

[48] Matlab (Feb. 2017).
URL www.mathworks.com/products/matlab/

[49] P. P. Neumann, V. Hernandez Bennetts, A. J. Lilienthal, M. Bartholmai, J. H. Schiller, Gas source localization with a micro-drone using bio-inspired and particle filter-based algorithms, Advanced Robotics 27 (9) (2013) 725–738.

[50] J. Rodriguez, C. Castiblanco, I. Mondragon, J. Colorado, Low-cost quadrotor applied for visual detection of landmine-like objects, in: Unmanned Aircraft Systems (ICUAS), 2014 International Conference on, IEEE, 2014, pp. 83–88.

[51] Trashout (May 2017).
URL https://www.trashout.ngo/trashmap/?ref=trashout.me

[52] See-demining (May 2017).
URL http://www.seedemining.org/main.htm.