Antonio L Alfeo, Mario G C A Cimino, Gigliola Vaglini, "Enhancing biologically inspired swarm behavior: metaheuristics to foster the optimization of UAVs coordination in target search", COMPUTERS & OPERATIONS RESEARCH, n. 110, pp. 34-47 (2019), doi: 10.1016/j.cor.2019.05.021

Enhancing biologically inspired swarm behavior: metaheuristics to foster the optimization of UAVs coordination in target search

Antonio L. Alfeo[a], Mario G. C. A. Cimino[a,*], Gigliola Vaglini[a]

[a]*Dept. of Information Engineering, University of Pisa, largo L. Lazzarino 1, Pisa, Italy.*

Abstract

Recent miniaturization in Unmanned Aerial Vehicles (UAVs) technology encourages the use of many small UAVs for search missions in unknown environments, provided that the autonomous and adaptive coordination logic can be effective. In this research field, biologically inspired metaheuristics have been proposed to mimics swarms, flocks, and other coordination schemas. The design and management of such systems is a research challenge when considering (i) combination and optimization of multiple metaheuristics and (ii) enhancements of biologically inspired metaheuristic through technological advances. In this paper the swarm coordination of UAVs employed in target search is based on flocking and stigmergy, to provide robust formation control and dynamic environmental information sharing, respectively. The design of both metaheuristics takes into account UAVs equipment, and the coordination logic is adapted to the mission by means of a differential evolutionary algorithm. This algorithm optimizes the aggregated structural parameters of all metaheuristics to allow the most efficient coordination with respect to the mission environment. Some possible enhancements of stigmergy are studied by simulating target search tasks on synthetic and real-world scenarios.

Keywords: UAVs, Swarm Intelligence, Stigmergy, Flocking, Differential Evolution, Distributed Targets Search

---

1. Introduction and motivation

Thanks to recent technological advancement, Unmanned Aerial Vehicles (UAVs) have become more and more accessible and are now increasingly utilized across a variety of domains, including exploration of unknown territories and monitoring of hazardous environments. Indeed, by remotely controlling a UAV,

*Corresponding author.
Email addresses: `luca.alfeo@ing.unipi.it` (Antonio L. Alfeo), `mario.cimino@unipi.it` (Mario G. C. A. Cimino), `gigliola.vaglini@unipi.it` (Gigliola Vaglini)

an operator on the ground can exploit the UAV sensing capabilities while addressing the so-called D3 missions ("dull, dirty and/or dangerous" (Whitehead et al., 2014)), where the access to humans is limited, dangerous or impossible. A number of considerations discourage the use of single UAV in favor of UAVs group (Whitehead et al., 2014). To use a unique UAV results in a single point of failure: an unpredictable hardware and/or software fault can prevent the mission completion (Suárez Fernández-Miranda et al., 2016). Moreover, the use of a single UAV demands high requirements in terms of reliability and durability, which usually means higher costs of design, construction, and maintenance. Considering the use of multiple UAV, a relevant issue is to reduce the coordination complexity demanded by centralized logic solutions (McCune et al., 2013). In a decentralized approach, a certain degree of autonomy can be exhibited by the UAVs swarm (Meng et al., 2014).

In its simplest form, a swarm of UAVs is characterized by a large number of homogeneous individuals (agents) with local communication, sensing and actuation capabilities (Maza et al., 2015). Such systems present different advantages: (i) it allows parallel/collective scan by exploiting the principles of self-organization, similarly to natural swarm; (ii) it is scalable, since by increasing the number of agents in the swarm its effectiveness is poorly compromised; (iii) it is flexible, in fact, due to the simplicity of a single agent logic, it can be adapted according to the scenario; (iv) it is robust, and then the task accomplishment is not affected by the fault of some swarm members (Schmickl et al., 2011) (Aznar et al., 2014).

The swarm behavior can be designed by employing three main conceptual frameworks (Brambilla et al., 2013): (i) *collective decision making*, in which UAVs aim at achieving consensus on a collective strategy, in order to maximize the performance of the swarm; (ii) *spatially organized behavior*, in which UAVs aggregate according to a set of spatial constraints to arrange themselves following specific patterns, chains, or structures; (iii) *navigation-based behavior*, in which each UAV follows a set of simple behavioral rules to steer the whole swarm and cooperatively explore an environment. The latter is mostly used for search and coverage task (Brambilla et al., 2013) and is frequently designed by exploiting biologically inspired approaches (Alfeo et al., 2018).

In this paper the swarm behavior is modeled considering and comparing two different paradigms, namely *biological behavior*, which mimics social animal metaheuristics, and *computational behavior*, which considers enhancements exploiting UAVs information technology. Our aim is to verify whether some enhancements allow a reduction of complexity and a more effective optimization of the structural parameters. Indeed, the digital environments can allow a specialization of biological models able to simplify both mechanisms and dynamics and then the research space. Nevertheless, this computational behavior keeps the essential benefits of the original biological behavior. Specifically, the biological behavior focuses on the following aspects:

(i) *biological stigmergy*: used to release an attractive/repulsive potential in areas where targets have been/have not been found (Sauter et al., 2005a).

An agent leaves a mark in the environment while performing an action, whereas other agents, by sensing the mark, are stimulated to perform an action. The mark dynamics is ruled by the *chemical diffusion* and *evaporation model*;

(ii) *biological flocking*: used to organize a flexible agents formation for collaborative sensing (Kaminka et al., 2008). The best known Reynolds model for flocking (Reynolds, 1987), is based on the rules of *alignment*, *separation*, and *cohesion*: alignment aligns the agents heading to the average heading of nearby agents (flock mates); separation prevents the overlapping of sensing areas by maintaining a minimum distance among flock mates; cohesion directs each agent towards the center of the flock mates;

(iii) *biological sensing/actuation*: target sensing occurs if the agent is over the target; obstacle sensing occurs if the obstacle is one step forward; pheromone release occurs at the same agents position; agent can be either still or moving (without incremental speed and rotation control). Finally, olfactory perception simply reflects the pheromone intensity (without overstimulation control).

The computational behavior characterizes the following aspects:

(i) *computational stigmergy*: since the digital pheromone is maintained in a virtual space, called pheromone map, it can have an instant diffusion, to immediately propagate the environmental information to nearby UAVs; furthermore, a *linear evaporation* and a *streamlined shape* allow a better control of the aggregated pheromone potential;

(ii) *computational flocking*: the simultaneous localization and mapping technology allows different-scaled flocking, ranging from visually-based flocks to large-scale localization-based formations, across physical barriers.

(iii) *computational sensing/actuation*: long-range sensing allows detection of remote targets and obstacles; as a consequence, computational pheromone can be released on a target remote with respect to the UAV position; moreover, a UAV can accelerate, decelerate and keep cruise speed while obstacles are not detected. Finally, olfactory receptors can decrease in sensibility over time to prevent overstimulation (olfactory habituation).

In this work, these behavioral models are exploited to solve the target search problem. According to (Senanayake et al., 2016) 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. It is worth noting that the mission duration does not include phases external to the search problem, such as the deployment time and the return time. 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. The proposed approach has been experimented in a proper simulation environment where synthetic and real-world target search missions are modelled.

Specifically, structural aspects of UAVs (such as speed, size, endurance, field of sight, collision avoidance, and so on) are fixed according to the type of UAV usually employed for each type of mission (land-mine search, gas leakage detection, illegal dumps discovery, etc.).

Externally to the simulator, a Differential Evolutionary (DE) algorithm controls the models aggregated parameters to search the most efficient coordination setting. The optimization process is an intrinsic part of the swarm: in some sense only after the optimization the initial set of UAVs becomes a swarm, i.e., an effective organism specialized for the type of mission.

The paper is structured as follows. Related works are presented in Section 2. In Section 3, the design of metaheuristics for UAVs coordination is detailed. Section 4 covers the design of missions and quality measures. Experimental results are presented and discussed in Section 5. Finally, Section 6 summarizes conclusions and future work.

2. Related work on swarm metaheuristics for target search

A lot of research has been developed in the field of metaheuristics for target search employing a swarm: this research is characterized by heterogeneity of goals, methodologies and scenarios. Some recent surveys (e.g., (Senanayake et al., 2016)) attempted to create a taxonomy discussing the qualitative differences between the proposed approaches. In this section some relevant works are briefly referred and summarized for the interested reader.

2.1. Stigmergy and Flocking metaheuristics

Among the biologically inspired approaches existing in the literature, stigmergy and flocking are widely used to coordinate a swarm of UAVs in target search tasks (Parunak et al., 2002).

Authors in (Sauter et al., 2005b) proposed a swarming schema based on stigmergy, in which virtual pheromones are deposited on a pheromone map and sensed by agents. Specifically, movements and action decisions are taken by the walker agents , whereas avatar agents are committed to making estimation in absence of sensor information. The schema is applied to many scenarios, including target acquisition. In (Brust et al., 2017) the authors design a multi-hop clustering procedure combined with stigmergy to provide an optimal solution to a set of objectives, among which target detection and tracking. Moreover, they propose a pheromones model including both attractive and repulsive pheromones, to mark detected targets and visited areas, respectively. Another application of this kind of pheromones model is proposed in (Atten et al., 2016), for similar missions.

Flocking was first proposed by Reynolds (Reynolds, 1987). An example is proposed in (Vásárhelyi et al., 2014) where the authors present a decentralized coordination strategy for UAVs based on flocking. Flocking is used to maintain UAVs in communication range and to coordinate themselves during their task. In (Hauert et al., 2011) a flock of multiple fixed-wing UAVs flying at a relatively large distance from each other is proposed. A flocking strategy is applied also in (Quintero et al., 2013) where UAVs autonomously navigate in a search area.

A number of research works have proposed enhancements for such biological metaheuristics (Bayındır, 2016). The problem is that an enhancement of swarm metaheuristic can make the UAVs behavior problem-dependent, i.e., it can transform the metaheuristic into a heuristic. In general, a heuristic is tailored to the mission at hand, to take full advantage of the particularities of the problem. As a consequence, an optimal solution cannot be provided for different circumstances. In fact, it is well known that any problem-dependent heuristic biases the system, and make it less scalable and less manageable. In contrast, metaheuristics are problem-independent techniques that do not take advantage of domain knowledge and can be used as general purpose problem solving tools.

For example, a scenario of land-mines search in a forest is characterized by numerous and small obstacles (trees), with single targeted cells (landmines). A good strategy for this type of mission is to increase the number of isolated UAVs, and to provide them with small stigmergy range. In contrast, a scenario of early wildfire detection, or illegal dump discovery in rural areas, are characterized by few obstacles and clustered targeted cells. For this type of mission, a good strategy is to create large and sparse flocks, with large stigmergy range.

A metaheuristic may accept a temporary deterioration of the solution, for a better exploration of the solution space getting to a near-to-global optimum. Obviously, a parametric tuning is necessary to adapt a metaheuristic to the problem at hand and a poor quality tuning may lead to unpredictable result. For example, stigmergy with large and persistent pheromone might attract too much agents leading to inefficient search. Flocking with large visibility radius might result in a very rigid formation. Since in multi-agent systems the interaction of the agents is not simply deducible from the properties of the components, an important technique to study the global properties and their level of predictability is simulation that can be used also when proposing a variant of a metaheuristic. The following works summarize some variants of flocking and stigmergy used for multi-UAVs systems.

In this context, in (Paradzik and İnce, 2016) with respect to biologically inspired pheromone model, which is usually attractive/repulsive, the authors encode specific information as pheromone flavors. In (De Benedetti et al., 2017) the self-organization of a flock of UAVs employed on an area coverage mission is achieved by allowing short-range communication among UAVs through gossiping. In (Qiu and Duan, 2017) the authors propose a coordination strategy based on pigeon flocking model. To avoid obstacles, the highest individuals in the flock hierarchy are informed about the obstacles position, providing a path planning strategy based on an artificial potential field.

2.2. Differential evolutionary optimization

The particularity of the approach proposed here is that the flocking and stigmergy metaheuristics make, together with the sensing/actuation, an integrated logic which is parameterized as a whole by a differential evolution optimization. It is worth noting that in this case the search space includes the space of the combined metaheuristics. In this space, stigmergy and flocking can be reinforced/weakened as well as enabled/disabled or simply modified. For example, flocking with visibility radius equals to zero disables the flocking itself. Similarly, stigmergy with mark radius equals to zero or evaporation rate equals to 1 means no stigmergy at all.

Thus, an adaptation process can search the best aggregation of metaheuristics according to the specific mission and performance measure, which can be tested in the simulated environment, exploring solutions which are normally impossible to consider by human design (Singh et al., 2017), (Bloembergen et al., 2015).

This is confirmed by research works such as (Cimino et al., 2015a) in which an adaptation mechanism tuning the swarm coordination is needed to fit the current scenario layout. Another example of adaptive coordination is proposed in (Labella et al., 2006) whose authors propose a control system for swarm robots involved in an object retrieval task. By changing the probability of switching between different individual statuses and strategies, the authors show the relevant improvements of efficiency of real and simulated robots. Authors in (Colby et al., 2015) present a control and coordination strategy for a team of UAVs in an environmental surveillance domain by co-evolving neural network controllers with different fitness evaluation functions.

In (Ramirez-Atencia et al., 2017) the authors present a Multi-UAV Cooperative Mission Planning, via a hybrid approach based on Multi-Objective Genetic Algorithm (MOGA) and Constraint Satisfaction Problem (CSP). The CSP is used when some constraints are partially fulfilled, whereas the MOGA is used if all constraint are fulfilled. The authors tested the effectiveness of the proposed approach over 16 mission scenarios, with different complexities and under different mission situations.

In the literature, the metaheuristics of stigmergy and flocking have been extensively adopted for solving computational problems that can be reduced to finding sub-optimal paths through a problem space. The problem space can be either a physical space (e.g. vehicle routing) or a logical space (e.g. scheduling). In our approach, we adopt stigmergy and flocking for coordinating the target search in the two-dimensional physical space. A different problem is the parametric optimization of stigmergy and flocking, which is carried out in a multi-dimensional logical space of parameters. For this kind of problem, many stochastic optimization algorithms, such as Genetic Algorithm (GA), Differential Evolution (DE), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) can be used. In the literature, a number of 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 DE algorithm, which is one of the most powerful stochastic real-parameter optimization algorithms (Das and Suganthan, 2011), (Agrawal and Kaur, 2018). Compared with other popular methods such as GA and PSO, DE has attracted more attention for continuous optimization problems for its simplicity, efficiency, and adaptivity (Kachitvichyanukul, 2012), (Cimino et al., 2015b), (Jones and Bouffet, 2007).

3. Design of environment and metaheuristics

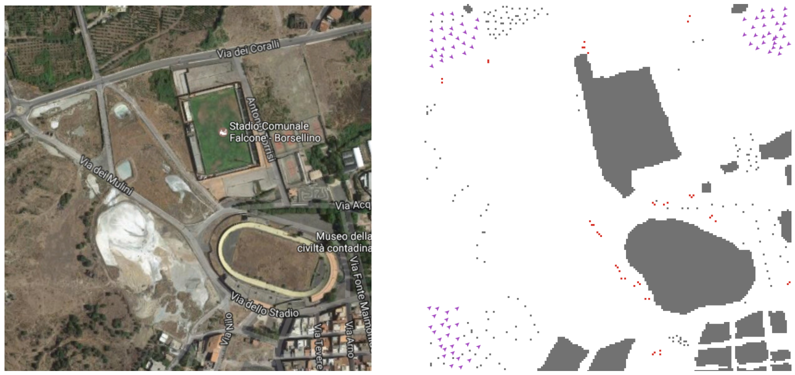
This section focuses on modeling the various aspects of UAVs coordination for a generic target search mission. Basically, the search environment contains UAVs, targets, and obstacles. The coordination logic of a UAV is based on collision avoidance, stigmergy, flocking and evolution.

3.1. Exploration simulator and basic collision avoidance

As discussed in the previous section, the design and the optimization of the coordination logic is highly supported by an exploration simulator. The first point is to distinguish between *flight* simulator and *exploration* simulator. A flight simulator focuses on control logic: it recreates the equations that govern UAV fly, how it reacts to external factors such as air density, turbulence, wind shear, cloud, precipitation, etc. In contrast, an exploration simulator focuses on coordination logic, assuming that external factors are already managed. It represents the exploration at a different scale, which depends on the spatial and temporal resolution needed to detect the target, and recreates at that scale obstacles and target distribution. Consequently, the basic drone movements and collision avoidance are simulated for the sole purpose of exploration. Thus, the drone acceleration (and deceleration) from zero to the cruise speed (and vice versa) is supposed to be linear assuming some conventional control logic and normal weather conditions. This occurs even when the exploration is made at small scales, such as for gas leak detection.

For this reason, the collision avoidance strategy is based on a standard approach (DJI, 2016), (DJI, 2017). Specifically, the UAV obstacle vision is set via two parameters, i.e., *ObstacleVision* and *ObstacleVisionAngle*, creating a circular sector area called *ObstacleVisionArea*, whose vertex is centered on the UAV. When an obstacle or another UAV is detected in the *ObstacleVisionArea*, the UAV changes its heading and speed to avoid the obstacle. With this model, the area that will be occupied by the UAV in the next instant can be easily calculated via its velocity and its heading. Thus, the multiple UAVs can be accordingly organized in the current instant so as to avoid overlapping with UAVs and obstacles in the next instant.

Requirements about velocity and heading information propagation can be fulfilled by current aerial networks technologies for target search applications (Hayat et al., 2016). Indeed, the coordination problem can be categorized as



(a) (b)

Figure 1: (a) Illegal Dump, satellite image (Google Maps), (b) Illegal Dump, vector image

delay-tolerant, since continuous connectivity is not a necessity, and a periodic data transfer (i.e., intermittent connectivity) is sufficient. On the other hand, since an exploration simulator focuses on coordination logic, it assumes that flight control factors are locally managed, by modeling the exploration problem at a large spatial-temporal scale. By assuming that the UAVs already have autonomous obstacle detection capability, it follows that Beyond-Line-of-Sight connectivity with the ground station and GPS positioning are adequate to control and intervene for any changes during the mission, and to get status updates (Hayat et al., 2016).

It is of the outmost importance to highlight that the layouts of interest are not arbitrary. They are rather based on some realistic applications of target search carried out by UAVs swarm, where maze-like layouts with possible deadlocks are not under consideration. For this reason, the coordination logic is not specifically designed for dealing with trap situations within the flight time of small UAVs.

3.2. Stigmergy metaheuristics

A single target detection activity corresponds to the identification of any part of it that is not already discovered. Specifically, the search problem is formulated by discretizing the environment into a set of cells. Each target is stationary and usually covers many cells. For example, in the case of a land mine a single targeted cell corresponds to a single target, but illegal dumps and gas leaks cover multiple targeted cells.

Fig. 1a and Fig. 1b show an example of abusive trash map in Paterno (Italy), and the corresponding vector image represented in the discretized environment, respectively (Cabreira et al., 2019). Here, obstacles (buildings and trees) are represented in dark gray, whereas targets are represented as red points. Drones, represented as purple triangles, are placed at the corners and are oriented towards the center of the area.

An efficient strategy in target search is to quick survey the area and to better explore only locations characterized by some circumstantial evidence (Kuyucu et al., 2015). This strategy requires that drones are dynamically arranged so as to be efficiently engaged when some member detects a targeted cell. For this purpose, a UAV detecting a targeted cell releases a virtual pheromone mark that acts as an attractive potential on neighboring UAVs, for a limited time established by evaporation.

In the biological behavior, the targeted cell is detected by an UAV only when such UAV is over it, and the mark is released at the same position. In contrast, in the computational behavior the UAV sensor has a range and angle of sensing, determining a remote detection; moreover, the pheromone mark can be released remotely with respect to the current UAV position (Fig. 2a). In the biological behavior, the pheromone is released on a single cell and diffused progressively on neighbor cells according to a chemical expansion model. Such expansion introduces transmission delay and complex dynamics that make the cooperation inefficient (Cimino et al., 2016). In the computational behavior, the pheromone deposit is instantly diffused, with a shape whose gradient is designed to attract UAVs towards a certain area around the target (Cimino et al., 2016).

As an example, Fig. 2b shows a UAV detecting pheromone within a circular area defined by *OlfactionRadius*. Such UAV adjusts its heading to the highest pheromone intensity. When a dense group of targeted cells is discovered, to allow an efficient UAV evacuation another metaheuristics is enabled: the *OlfactoryHabituation*, the time after which the UAV is temporarily unable to perceive pheromone (Fig. 2c).

According to the above requirements, the pheromone mark is modeled as in Fig. 2d. Its shape is a truncated cone determined by *TopRadius*, *BottomRadius*, and *markIntensity* (height). Pheromone marks can aggregate in pheromone tracks. The track evaporates over the time: every tick, its intensity is linearly reduced by a given amount (*evaporationRate*). Formally, the track intensity at the instant k can be described by Formula 1:

$$Track(k) = Track(k-1) + Mark(k) - \{evaporationRate \cdot markIntensity\} \quad (1)$$

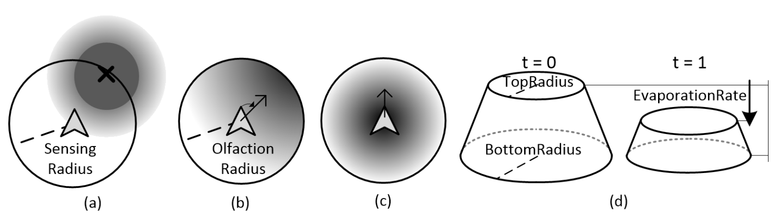


Figure 2: Computational Stigmergy. Pheromone Release (a), Pheromone sensing/attraction (b), Olfactory habituation (c), Pheromone Mark Implementation (d)

3.3. Flocking metaheuristics

Flocking behavior is based on a circular sector shaped sensing area represented in Fig. 3, and divided into three subareas as in Fig. 3a. Specifically, (i) if one or more UAVs (flockmates) are within the Separate Area (defined by *FlockAngle* and *SeparationRadius*), the UAV turns away from their barycenter by (at most) *MaxSeparationTurn* (Fig. 3b). The separation reduces the overlapping of UAVs sensing areas which causes a less efficient search. (ii) if the separate area is free, and some flockmates are in the strip between *SeparationRadius* and *AlignRadius* (Fig. 3c), the UAV turns towards the average heading of the flockmates, by at most *MaxAlignTurn*. (iii) if both the separate and the align areas are free, and some flockmates are in the strip between *AlignRadius* and *CohereRadius*, the UAV turns to point toward their barycenter, by at most *MaxCohereTurn* (Fig. 3d). This procedure can provide a sensible improvement to the search strategy in (Cimino et al., 2016), by reducing the number of isolated UAVs when the collaborative search is convenient. Indeed, an isolated UAV performs a random walk characterized by small random turns of at most *MaxRandomTurn*. Specifically, an agent is considered isolated if no other agents are located within the cohere radius. An example of communication model for this purpose is based on Beyond-Line-of-Sight communication with the ground station and global positioning, to get status updates. In the example, it is assumed that the agent has autonomous obstacle detection and relatively long range sensing capabilities. Thus, the flocking problem can be categorized as delay-tolerant, continuous connectivity is not a necessity, and a periodic data transfer (via intermittent connectivity) is sufficient for flocking (Hayat et al., 2016).

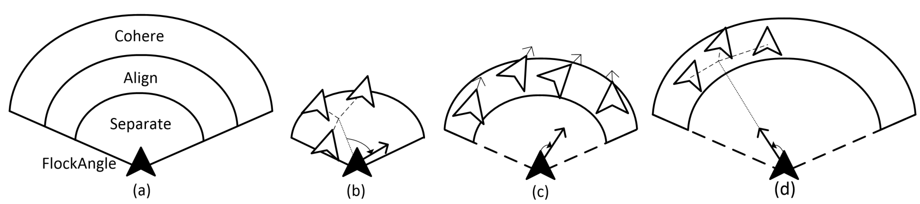


Figure 3: Computational Flocking. Flocking Procedures Areas (a), Separate (b), Align (c), Cohere (d).

3.4. Metaheuristics aggregation and its evolution

Every tick, a UAV executes the logic represented in Fig. 4. Here, four vertical lanes are shown: input parameter, coordination logic, actuation logic and output parameter. On the other side, the five horizontal lanes are: target detection, pheromone attraction, flocking, collision avoidance and finally movement.

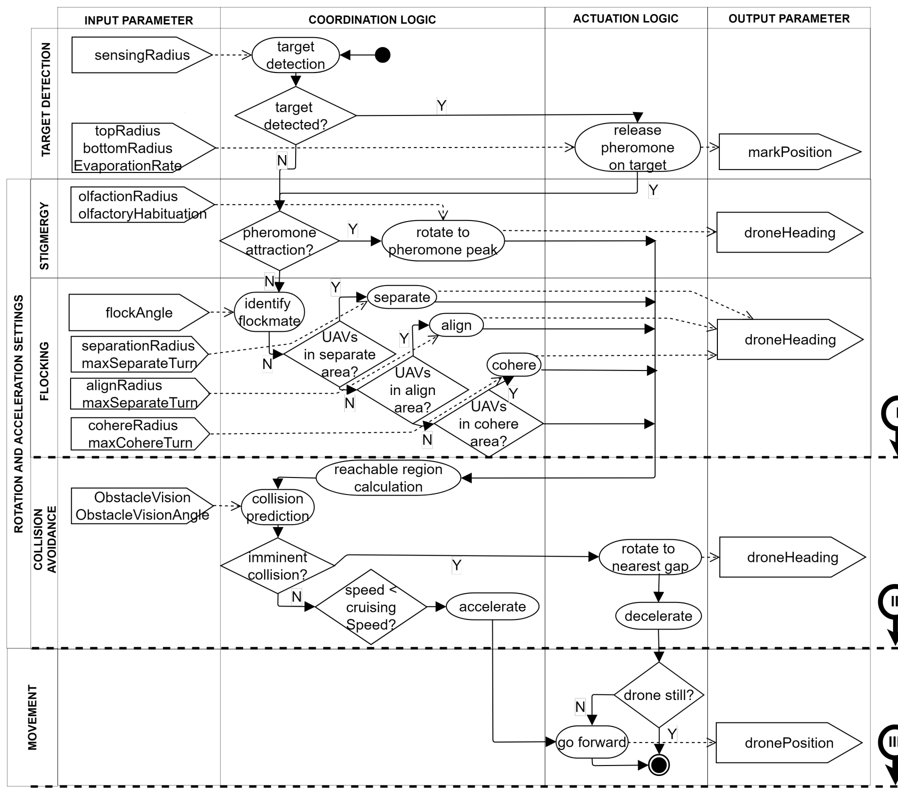


Figure 4: The metaheuristics aggregation carried out by a UAV.

Basically, the UAV checks whether the cell is targeted, and in such case releases a pheromone deposit. Then it tries to detect the existing pheromone and, if missing, it tries to detect flock mates (phase I). In case of detection of pheromone or flock mates, the heading is modified accordingly. Otherwise, phase II is executed, consisting in collision avoidance rules. Finally, one step movement is made (phase III). As a consequence, (i) the collision prediction, when occurring, cancels the effects of other rules, and (ii) the pheromone detection, when occurring, does not allow the execution of flocking rules.

As an example, Fig. 6 shows a simulation snapshot with examples of flocks, targets and pheromone. Fig. 4 shows some significant snapshots of a target search process. Here, after the first targeted cells discovered, the pheromone increases its intensity thanks to other attracted UAVs, allowing the incremental discovery of other targeted cells.

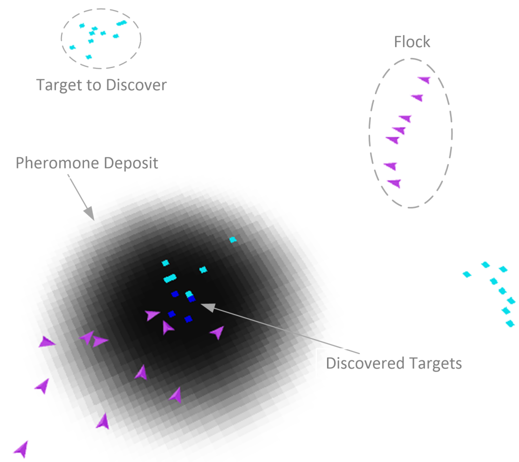


Figure 5: A simulation snapshot with flocks, targets and pheromone deposit.

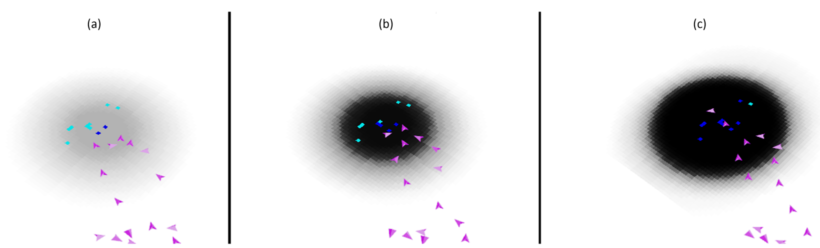


Figure 6: Some significant snapshots of target search. Tick: 10; targeted cells discovered: 2 (a). Tick: 14; targeted cells discovered: 4 (b). Tick: 18; targeted cells discovered: 9.

Table 1 summarizes the aggregated parameters of flocking and stigmergy. To find the best parameters that minimizes the duration of a specific mission is an optimization task. For this purpose, we exploit an evolutionary algorithm (Cimino et al., 2016). Among the different evolutionary algorithms, Differential Evolution (DE) exhibits excellent performances both in unimodal, multi-modal, separable, and non-separable problems. Moreover, DE exhibits both simplicity and efficiency in global optimization over continuous spaces (Das and Suganthan, 2011). As a performance measure, the mission duration is considered. Indeed, to reduce the flight time is the critical aspect of this kind of missions, due to the limited duration of the UAVs batteries.

The performance evaluation of a single solution involves a simulation of the entire mission. In addition, some level of indeterminism is contained in results, due to the level of randomization in the UAV turning. Thus, each evaluation is carried out three times (repeated trials). To sensibly reduce the computation time of the optimization, without scarifying a lot of accuracy, the mission goal can be considered achieved when the 95% of the targets has been found. For all the above reasons, for a given set of parameters, the average mission achievement time is calculated over 3 repeated simulations in order to determine the fitness value.

Table 1: Stigmergy and Flocking parameters.

Procedure	**Parameter Name**	**Measure**
Stigmergy	MarkTopRadius	*cells*
Stigmergy	MarkBottomRadius	*cells*
Stigmergy	MarkIntensity	*unit*
Stigmergy	EvaporationRate	*ratio*
Stigmergy	OlfactoryHabituationTime	*ticks*
Stigmergy	OlfactionRadius	*cells*
Flocking	FlockAngle	*degree*
Flocking	SeparationRadius	*cells*
Flocking	MaxSeparationTurn	*degree*
Flocking	AlignRadius	*cells*
Flocking	MaxAlignTurn	*degree*
Flocking	CohereRadius	*cells*
Flocking	MaxCohereTurn	*degree*
Flocking	MaxRandomTurn	*degree*

3.5. Evolutionary optimization of the aggregated metaheuristics

The DE variant adopted in this work is summarized by the pseudocode presented in Algorithm 1. More formally, in a simulated scenario (or mission), the swarm S_i explores an environment where $Obstacles$ and $Targets$ are statically specified. $AttractivePheromones$ and $RepulsivePheromones$ are dynamic structures released when an UAV is, and respectively, is not, positioned on an unknown target. Let K be the number of aggregated parameters. In DE, S_i is a solution represented by a real K-dimensional vector called genotype $\mathbf{p}_i$. The search time returned by the simulated mission is used as a fitness of the solution, f_i. The DE algorithm has at least two hyper-parameters: 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 2 and Algorithm 3 define the mutation and the crossover operators, respectively.

Specifically, DE starts with a population $P^{(0)}$, made by N candidate solutions, $\mathbf{p}_i^{(0)}$, randomly generated under user-specified parametric constraints. At each iteration t, and for each genotype $\mathbf{p}_i^{(t)}$ of the current population $P^{(t)}$ a mutant vector $\mathbf{m}$ is created by applying the mutation of randomly selected members. Then, a trial vector $\mathbf{p}_i^*$ is created by crossover of $\mathbf{m}$ and $\mathbf{p}_i^{(t)}$. Subsequently, the population is modified selecting 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, i.e., number of iterations performed or adequate fitness reached (Das et al., 2016), the vector characterizing the swarm with the best fitness (i.e. the shortest search time) in the current population is considered as the optimal swarm parameterization.

Algorithm 1: Differential Evolution Algorithm

```
function DifferentialEvolution(Obstacles, Targets, AttractivePheromones,
  RepulsivePheromones)
t = 0;
P^(0) = initializePopulation();
for each genotype p_i^(0) in P^(0) do
    S_i^(0) = genotypeToSwarm(p_i^(0));
    f_i^(0) = simulateMission(S_i^(0), Obstacles, Targets, AttractivePheromones,
      RepulsivePheromones);
do
    for each genotype p_i^(t) in P^(t) do
        m = generateMutant(P^(t), p_i^(t));
        p_i^* = binomialCrossover(p_i^(t), m);
        S_i^* = genotypeToSwarm(p_i^*);
        f_i^* = simulateMission(S_i^*, Obstacles, Targets, AttractivePheromones,
          RepulsivePheromones);
    for each genotype p_i^(t) in P^(t) do
        if (f_i^* < f_i^(t)) then
            p_i^(t+1) = p_i^*; f_i^(t+1) = f_i^*;
        else
            p_i^(t+1) = 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(p_min^(t));
```

Algorithm 2: Mutant vector generation

```
function generateMutant(P^(t), p_i^(t))
p' = randomExtraction(P^(t) \{p_i^(t)});
p'' = randomExtraction(P^(t) \{p_i^(t), p'});
p''' = randomExtraction(P^(t) \{p_i^(t), p', p''});
return p' + F· (p'' - p''');
```

Algorithm 3: Binomial crossover

```
function binomialCrossover(p_i^(t), m)
k = randomInteger(1, K);
for each j-th gene p_{j,i}^(t) in 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 w;
```

4. Considered scenarios and quality measures

In this section, the scenarios used and the various quality measurements are presented and motivated.

4.1. Simulation environment and scenarios

The simulation environment features a virtual environment representing the search field. In the design of the testbed, an important distinction is between *flight* simulator and *exploration* simulator. A flight simulator focuses on control logic: it recreates the equations that govern UAV fly, how it reacts to external factors such as air density, turbulence, wind shear, cloud, precipitation, etc. In contrast, an exploration simulator focuses on coordination logic, assuming that external factors are already managed. It represents the exploration at a different scale, which depends on the spatial and temporal resolution needed to detect the target, and recreates at that scale obstacles and target distribution. Consequently, in the environment the basic UAV movements and collision avoidance are simulated for the specific purpose of exploration.

Fig. 7 shows a simplified representation of the environment with the available elements. For a better granularity and without loss of generality, the search problem is formulated by discretizing the environment into a lattice of cells. In the environment, a single UAV, or drone, is represented by a disc with an inner arrowhead. An obstacle or a target usually covers many cells. In figure, each obstacle-cell is black, whereas each targeted cell is denoted by ”x”. Finally, a pheromone mark is represented as a cluster of grey cells. The grey level represents the pheromone intensity.

Obstacle and target cells are static. Every tick, the environment changes its current state to the next state, according to the metaheuristics aggregated rules: (i) the grey level of a pheromone cell is dynamic, and it is updated following the evaporation rule; (ii) an attractive or repulsive pheromone mark is released by a drone according to the target detection rules; (iii) the drone position and direction is dynamic and set according to the stigmergy, flocking and collision avoidance rules. The environment and the coordination logic are implemented on NetLogo, a leading simulation platform for swarm intelligence (Netlogo, 2017). For a given type of mission, all rules are parametrically optimized by the DE adaptation subsystem developed on Matlab, a numerical

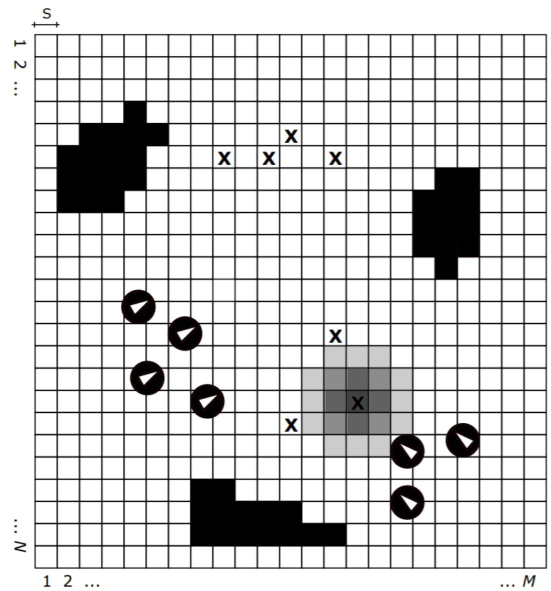


Figure 7: Environment and its elements: drones (circles), target (cells denoted by x), pheromone (grey cell), and obstacle (black cells).

optimization framework (Mathworks, 2018). The source code of the integrated system, together with the scenarios, has been developed in the framework of an institutionally funded research project, and publicly released on the Github platform (Cimino et al., 2018).

For each simulated scenario, one patch corresponds to a squared area of 1 meters length in the corresponding real-world scenario. Fig. 8 shows the considered scenarios.

Three of the scenarios are synthetic: *Field*: an open field with some clustered targeted cells; *Forest*: a forest in search and rescue operation; *Urban*: a urban environment with the gas leakage. Other three real-world scenarios are derived from publicly available data set: *Dump*, map of illegal dumps in Patern, Sicily (Italy); *Rural Mine* and *Urban Mine*, map of unexploded land-mines near Sarajevo, Bosnia-Herzegovina. Each scenario is explored by 80 UAVs, which are initially placed at the corners of the map. The characteristics of each scenario are detailed in Table 2.

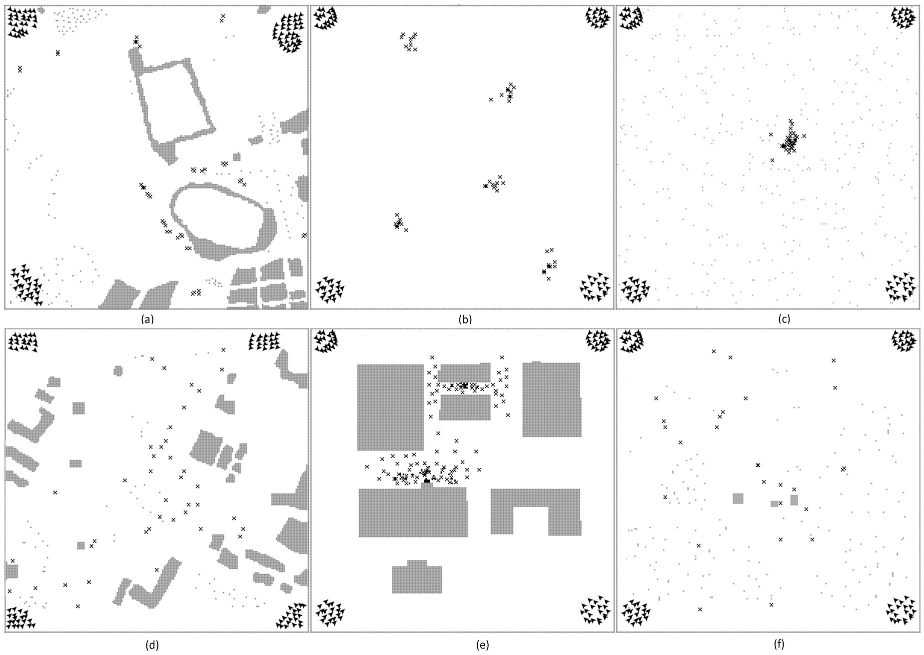


Figure 8: Scenarios: (a) Dump. (b) Field. (c) Forest. (d) Rural Mine. (e) Urban. (f) Urban Mine.

Table 2: Scenarios characteristics.

Scenario	Source	Obstacles	Buildings	Targets
Dump	(Trashout, 2018)	140	19	42
Field	Synthetic	0	0	50
Forest	Synthetic	400	0	20
Rural Mine	(See-demining, 2018)	281	3	28
Urban	Synthetic	0	7	112
Urban Mine	(See-demining, 2018)	59	28	40

Each temporal tick of the simulation has been set to represent one second in the real world search tasks. The simulator takes into account UAVs cruise speed, acceleration, angular velocity, battery duration, UAV size, and sensing. These characteristics have been set considering the technical specifications of the UAV possibly used in the scenario. Specifically, for the scenarios Dump, Field and Forest, the UAV used is DJI Inspire 2 with Zenmuse X5S video camera (Cruz et al., 2016), *DJI2+X5S vc* for short. For the scenarios Rural Mine, Urban and Urban Mine, the UAV is equipped to detect gas and landmines (Gade and Moeslund, 2014) such as the DJI Inspire 1 with Zenmuse XT thermal camera (Chiaraviglio et al., 2016), *DJI1+XT tc* for short. Table 3 summarizes all structural parameters.

Let us consider the UAV camera pointing downwards. It is possible to

Table 3: Structural parameters of the UAV models.

Parameter Name	**Measure**	***DJI2+X5S vc***	***DJI1+XT tc***
Max Speed	m/s	26	22
Cruising speed	m/s	3	3
Max Acceleration	m/s^2	4	4
Max Angular velocity	$degree/s$	150	150
Endurance time	s	1620	1080
UAV size	m	0.6	0.6
Sensing radius	m	2	4
ObstacleVision	m	3	3
ObstacleVisionAngle	$degree$	15	15

calculate the radius of a circular sensing area (*Sensing Radius*) considering a UAV flight altitude of 3 meters, to navigate under the canopy or in a cluttered environment (Israel, 2011). A UAV cruising speed no higher than 3 m/s is assumed to ensure a good target recognition performance (Rodriguez et al., 2014).

Among the DE variants available in the literature, the popular one called $DE/rand/1/bin$ has been adopted (Das and Suganthan, 2011), 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. Among the available variants, the $DE/rand/1/bin$ provides diversity, promoting the exploration of the search space (Das et al., 2016).

The *cross-over probability* and *differential weight*, have been set up to 0.5 and 0.7, respectively. Finally, the *MarkIntensity* is set to 1, since any stigmergic behavior can be managed directly by modifying the *evaporationRate*.

4.2. Quality measurements

In the literature, it is well-known that to provide a general purpose quality measure of the swarm properties is a challenging task (Birdsey et al., 2017). However, the designer can identify the most convenient properties for each specific application, and evaluate how much the swarm meets them. For this reason, we propose a set of quality measures to investigate the performance and the properties of the proposed approach:

(i) *Effectiveness*: since battery duration is a critical feature of current UAVs, a measure of the mission effectiveness is the average completion time, i.e. the average time to find the 95% of the targets.

(ii) *Efficiency*: the efficient swarm carries out a quick survey of the scenario (exploration) and an in-depth search of areas characterized by targeted cells (exploitation). The trade-off between exploration and exploitation is the target search efficiency. Formally, let $v_{(x,y)}$ be the number of visits that the cell (x, y) of the environment has received during the mission, and $d_{t(x,y)}$ the distance of (x, y) from the closest target. Thus, we expect that for an efficient search $v_{(x,y)}$ is large/small for cells that are close to/far from a target. In other terms, the average ratio between $v_{(x,y)}$ and $d_{t(x,y)}$ is high:

$$SearchEfficiency = \frac{1}{n} \cdot \sum_{x=1}^{\sqrt{n}} \sum_{y=1}^{\sqrt{n}} \frac{v_{(x,y)}}{d_{t(x,y)}} \tag{2}$$

(iii) *Scalability*: By increasing the number of UAVs the search time is expected to be lower, due to the additional resources. However, the search time can be negatively affected by the number of path deviations caused by collision avoidance between many UAVs. The average search time over repeated trials, calculated for increasing numbers of UAVs, provides a scalability measure.

(iv) *Adaptive cooperation*: on average it is expected that: (a) stigmergy attracts mainly flock mates, i.e., the average flock size and the pheromone size should be similar; (b) a scenario with isolated/aggregated targeted cells determines small/large flocks; (c) obstacles determine flock fragmentation. To measure such effects, the following measures should be compared: (i) the (average) width of flocks, i.e. the (average) maximum distance between two flock mates; (ii) the average number of isolated UAVs, i.e. UAVs not belonging to any flock; (iii) the average number of non-flock mates attracted by a pheromone deposit (*flock dynamism*). Such measures are based on a procedure able to dynamically identify the flock composition. For this purpose, we adopt a well-known density-based clustering algorithm, DBSCAN (Loh and Park, 2014). Fig. 9 shows the DBSCAN logic: an agent p is a *coreagent* if at least $minPts$ agents are within distance ϵ of it. Such agents are said to be directly reachable from p. An agent q is reachable from p if there is a path $p_1, ..., p_n$ with $p_1 = p$ and $p_n = q$, where each p_{i+1} is directly reachable from p_i and all the agents on the path are core agents. If p is a core agent, then it forms a flock together with all agents that are reachable from it.

By exploiting the maximum distance of flockmate's interaction (i.e. CohereRadius) as ϵ we are able to generate clusters that correspond to the arrangement of the UAVs in the flocks. Specifically, a group of directly reachable UAVs are considered a flock, while the ones classified as noise are considered single UAVs (i.e. not belonging to any flock).

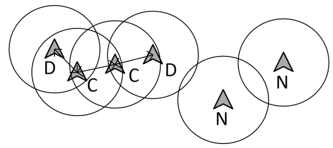


Figure 9: Representation of the density-based flock composition procedure: Core agents (C), Directly Reachable Agents (D), and isolated agents (N).

5. Experimental results

This section assesses and discusses the properties of the proposed algorithms for each scenario, by using the quality measures presented in the former section. For each measure, we present the 95% Confidence Interval (CI) over 5 repeated trials. The best results are highlighted in bold style.

Table 4 shows the mission duration for different computational *sensing radiuses* (abbreviated as *Sens.Radius* in the tables), assuming the computational actuation, i.e., remote sensing and remote pheromone release. Here, the enhancement provided by computational sensing and actuation is apparent. Table 5 clearly shows the same experiment assuming the biological actuation, i.e., remote sensing and local pheromone release. Comparing Table 4 and Table 5 it is apparent that the complete computational sensing and actuation sensibly improves performance.

Table 4: Mission duration with computational sensing and actuation.

Scenario	Sens.Radius 2	Sens.Radius 6	Sens.Radius 15
Dump	363.2 ± 102.6	238.4 ± 115.9	**108.6 ± 42.8**
Field	115.4 ± 45.7	52.8 ± 7.5	**35.4 ± 1.41**
Forest	334.8 ± 73.3	181.2 ± 88.0	**123.2 ± 25.04**
Rural Mine	195.8 ± 49.6	111.8 ± 68.5	**38.6 ± 2.42**
Urban	801.6 ± 310.7	727.8 ± 511.8	**286.2 ± 179.1**
Urban Mine	303 ± 85.7	160 ± 43.89	**100.2 ± 23.46**

Table 5: Mission duration with computational sensing.

Scenario	Sens.Radius 2	Sens.Radius 6	Sens.Radius 15
Dump	440.6 ± 192.58	**234.2 ± 129.8**	245.2 ± 91.4
Field	139.8 ± 66.4	61.2 ± 15.9	**49.2 ± 11.59**
Forest	372.4 ± 66.2	246.4 ± 73.2	**186.6 ± 65.2**
Rural Mine	193 ± 33.8	94.8 ± 23.28	**40.6 ± 1.11**
Urban	1070.8 ± 393.2	**722.6 ± 360.0**	942 ± 695.8
Urban Mine	363.8 ± 134.5	195.4 ± 35.2	**129.4 ± 37.31**

Based on these results, the computational behavior is compared with the biological behavior based only on adaptive Stigmergy and Flocking, called *S+F** presented in (Cimino et al., 2016). It is worth noting that the *S+F** algorithm supports a basic obstacle avoidance that does not detect drones as obstacles. Thus, UAVs overlapping are possible. To have comparable results, the same UAV parameters reported in (Cimino et al., 2016) have been used. The results in Tab. 6, clearly show that the computational approach outperforms the *S+F** approach, although the latter is not constrained by UAV avoidance. Indeed, in the Urban scenario, which is characterized by highly dense targeted cells, the *S+F** approach outperforms the computational approach. However, it is worth noting that the number of UAV overlapping during the execution of the *S+F** approach on this scenario is very high. Thus, the related result is not realistic since it does not consider the real-world constraints by providing an excess of freedom to UAV flight.

Table 6: Mission duration with computational approach and *S+F** approach.

Scenario	***S+F****	**Computational Approach**
Dump	**927 ± 71**	919.9 ± 180.9
Field	500 ± 55	**292 ± 41.9**
Forest	515 ± 65	**511.3 ± 38.9**
Rural Mine	947 ± 57	**501 ± 95.3**
Urban	**644 ± 51**	930.4 ± 114.1
Urban Mine	1109 ± 67	902.6 ± 133.2

Table 7 and Table 8 show the mission duration and the search efficiency when considering the adaptation of the evolution metaheuristics. Here, the human adaptation is provided by means of heuristics based on simple statistics: (i) *MaxRandomTurn* and *FlockAngle* are set to 120 and 60 degrees, respectively; (ii) to avoid overlapping the UAVs fields of view, *SeparationRadius* is set to 2 cells, whereas *AlignRadius* and *CohereRadius* are set to 4 and 8 cells, respectively; (iii) the flocking parameters, *SeparationRadius*, *AlignRadius* and *CohereRadius*, have been set accordingly to 10, 15 and 20 cells; (iv) *MarkTopRadius* has been set to 6 cells, since it must be greater than *ObstacleVision* to avoid UAV overcrowding; (v) *MarkBottomRadius* has been set to 8 cells, since this is the average distance between targeted cells in the scenarios; (vi) *OlfactoryHabituationTime* is set to 3 ticks to allow the UAV to cover a distance of *markBottomRadius* which allows to go far from its pheromone.

Table 7: Mission duration considering the adaptation process performed by DE.

Scenario	**Random Walk**	**Human Adaptation**	**DE Adaptation**
Dump	701.8 ± 87.8	624.4 ± 382.7	**419.4 ± 157.0**
Field	236.6 ± 14.2	198.6 ± 43.3	**119.2 ± 16.8**
Forest	532.2 ± 130.1	602.2 ± 171.1	**318.2 ± 26.4**
Rural Mine	206.0 ± 25.4	200 ± 31.4	**110.4 ± 58.9**
Urban	1760.2 ± 398.6	3056.2 ± 1919.8	**801.6 ± 310.7**
Urban Mine	217.2 ± 56.7	**200.6 ± 17.7**	198.2 ± 35.2

Table 8: Search Efficiency considering the adaptation process performed by DE.

Scenario	Random Walk	Human Adaptation	DE Adaptation
Dump	0.0838 ± 0.004	0.0737 ± 0.004	$\mathbf{0.0652 \pm 0.002}$
Field	0.0680 ± 0.057	0.0194 ± 0.001	$\mathbf{0.0156 \pm 0.001}$
Forest	0.0229 ± 0.002	0.0247 ± 0.002	$\mathbf{0.0138 \pm 0.001}$
Rural Mine	0.0274 ± 0.001	0.0269 ± 0.002	$\mathbf{0.0264 \pm 0.001}$
Urban	0.5294 ± 0.175	0.6630 ± 0.169	$\mathbf{0.1937 \pm 0.051}$
Urban Mine	0.0270 ± 0.002	0.0262 ± 0.002	$\mathbf{0.0259 \pm 0.001}$

The results obtained with the DE adaptation shows a good improvement of performances in all scenarios, with respect to the other approaches. It is worth noting that in many scenarios the Human Adaptation does not outperforms the Random Walk. This result shows the structural importance of the DE optimization.

The better search efficiency of the DE adaptation is also confirmed by Fig. 10, where the distribution of number of cells with a given number of visits is shown. Here, an efficient search produces narrower and more-to-the-left distributions.

To measure the internal UAV organization caused by the DE adaptation, Table 9 and Table 10 shows the adaptive cooperation measures with Human and DE adaptation, respectively. As a first result, in the DE adaptation, each scenario is characterized by a lower number of isolated UAV with respect to the Human adaptation. This corresponds to a better swarm formation.

Table 9: Adaptive cooperation measures with human adaptation.

Scenario	Flock Size	Isolated UAVs	Flock Dynamism
Dump	28.463 ± 0.215	47.539 ± 0.347	0.970 ± 0.182
Field	24.473 ± 0.293	47.084 ± 1.119	0.983 ± 0.193
Forest	25.788 ± 0.198	51.196 ± 0.395	1.396 ± 0.280
Rural Mine	25.155 ± 0.343	47.830 ± 0.806	1.437 ± 0.231
Urban	25.899 ± 0.109	50.593 ± 0.129	1.333 ± 0.178
Urban Mine	29.326 ± 0.451	42.098 ± 0.757	1.000 ± 0.179

Table 10: Adaptive cooperation measures with DE adaptation.

Scenario	Flock Size	Isolated UAVs	Flock Dynamism
Dump	34.334 ± 0.350	5.104 ± 0.119	0.190 ± 0.070
Field	9.213 ± 0.365	35.721 ± 1.236	3.033 ± 0.367
Forest	24.901 ± 0.261	11.114 ± 0.214	0.842 ± 0.247
Rural Mine	16.816 ± 0.361	17.922 ± 0.498	0.148 ± 0.067
Urban	9.875 ± 0.076	29.726 ± 0.221	2.041 ± 0.234
Urban Mine	29.964 ± 0.488	7.910 ± 0.229	0.858 ± 0.174

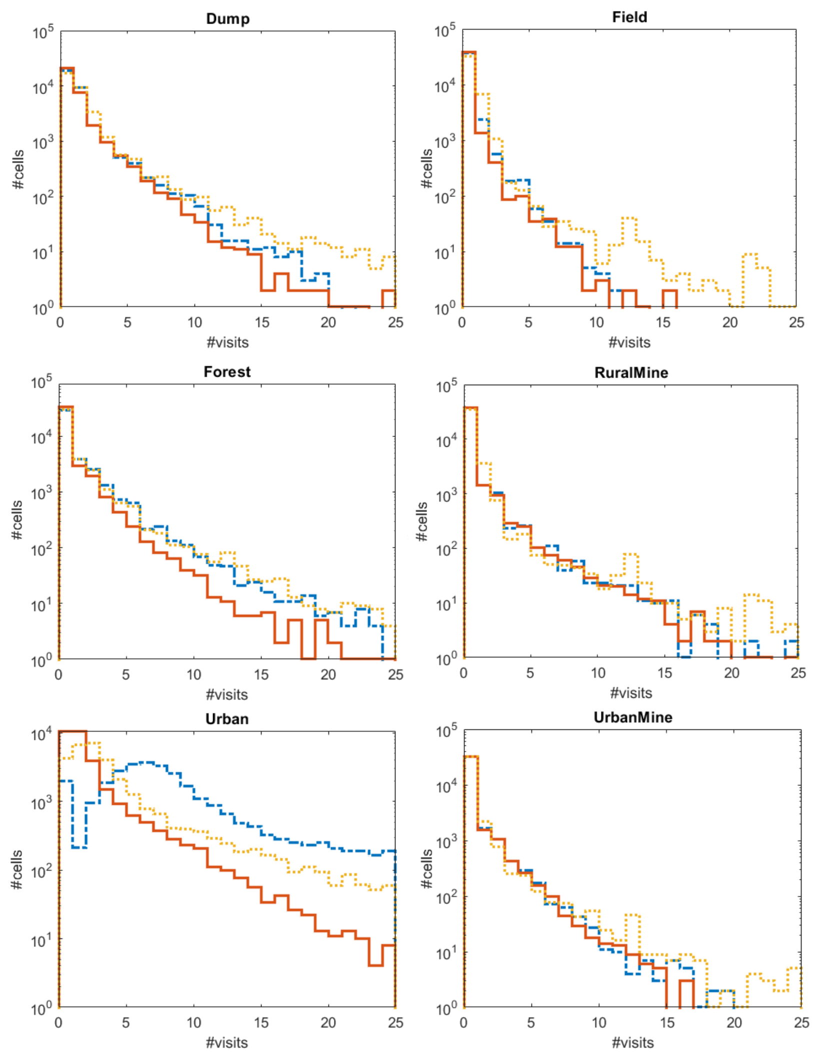


Figure 10: Search efficiency: number of cells against number of visits, with Random Walk (dotted line), Human Adaptation (dashed line), and DE adaptation (bold line).

On the other hand, the DE adaptation determines flock configurations that are peculiar to the scenario structure. For example, Dump, Forest and Urban Mine scenarios are characterized by numerous and small obstacles, which tend to crumble dense and rigid flocks. For this purpose, a good strategy is to create large and sparse flocks, which are more resilient to small obstacles getting across. This also results in a lower number of isolated UAVs shown in Table 10.

Figure 11 shows the scalability of the proposed approach, considering the confidence intervals of mission duration against the number of drones. Overall it shows that for each scenario, to increment the number of drones produces a different degree of improvements. It is worth noting that for some scenario characterized by dense targeted cells or dense obstacles such as Urban, Field, and Rural Mine, some nonlinear phenomenon can occur due to the complex obstacle avoidance situations.

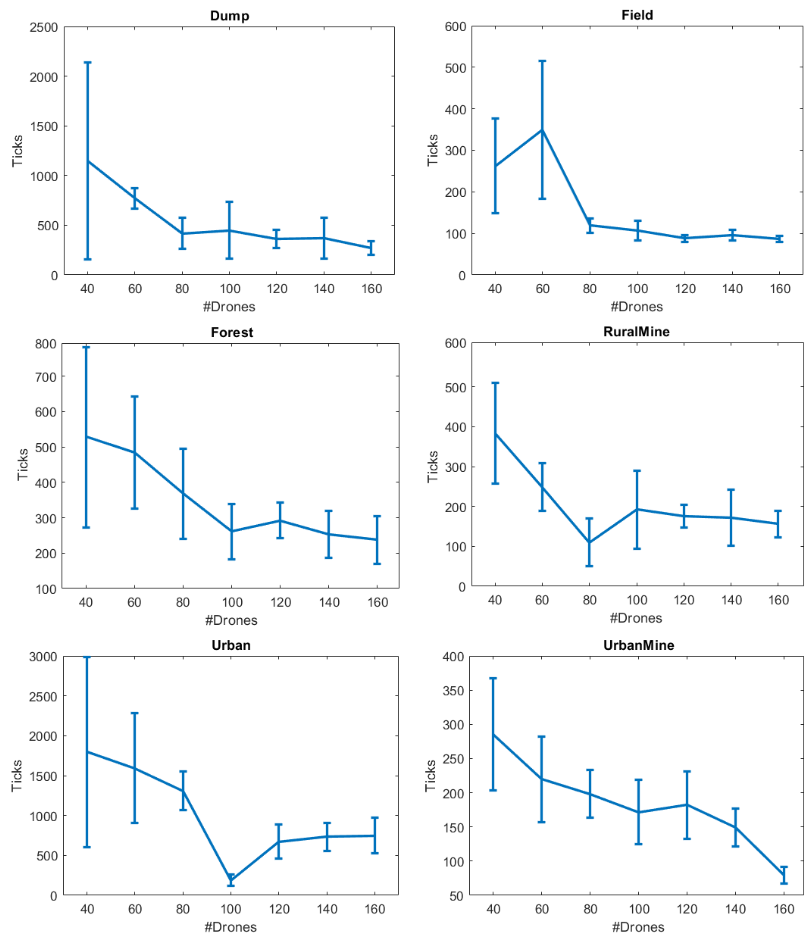


Figure 11: Mission duration with the computational approach: confidence intervals against UAVs numbers.

The runtime of the DE depends linearly on the population size (let us say, P) and on the maximum number of generations (G). In the worst case scenario, the simulated search mission is repeated $G \cdot P$ times. The runtime of each simulated search mission depends on the complexity of the scenario (e.g. size, number of targets, the arrangement of the obstacles) and on the effectiveness of the coordination mechanism. Such characteristics are difficult to formalize, however, we can provide an order of magnitude of the runtime by measuring a confidence interval for each scenario. Table 11 shows the runtime values in hours, calculated for each scenario, as the 95% confidence interval over 5 runs. The maximum number of generations G has been set to 15, and the population P has been set to 30. The following hardware/software platform was used: CPU Intel Core i7-6700HQ at 2.60-3.50 GHz, 6M Cache, 16 GB DDR3L 1600MHz RAM, Windows 10 OS. Considering that the code is not engineered for parallel computing, and then the evaluation of each individual is sequential, the resulting adaptation is relatively fast.

Table 11: Adaptation time (hours).

Dump	**Field**	**Forest**	**RuralMine**	**Urban**	**UrbanMine**
10.1 ± 1.3	4.9 ± 0.3	17.4 ± 0.9	4.5 ± 0.3	18.2 ± 1.0	7.3 ± 0.5

6. Conclusion and future work

In this work, we have enhanced basic biologically inspired metaheuristics for modeling and optimizing UAVs coordination in target search, taking into account technological and computational advances. A novel coordination logic is proposed, by designing stigmergy, flocking, and sensing/actuation metaheuristics. The combined metaheuristics are logically integrated and adapted to specific missions via a differential evolution optimization. The system has been simulated on synthetic and real-world scenarios, by considering UAV commercial models. Experimental results show that (i) the computational advances sensibly improve the performance and the realism of the biologically metaheuristics; (ii) the differential evolution optimization provides significant and structural improvements to the coordination logic; (iii) the quality of the resulting cooperation is better in terms of swarm formation, search efficiency, strategy and scalability.

In the current system, the constrained optimization is modelled by setting hard constraints, i.e., intervals for each parameter; nevertheless, it might be interesting, for example, to penalize some ranges of parameters values in the objective function. The so called soft constraints allow to distinguish the various solutions in terms of technological cost, to trade-off between efficiency and cost. On the other hand, in this research a strong emphasis on reducing the flight time has been given in the optimization process. This emphasis creates efficient swarms, but, for example, given an industrial plant, the user might be interested in a simpler coordination strategy working sub optimally but useful in similar layouts beyond the specific plant. Thus, in this case, the system should be configurable so as to reduce the precision and increase the simplicity. For these reasons, future work will focus on the design of an objective function including soft constraints and exploiting knowledge on the environment during the swarm optimization.

The robustness to faults and loss of individuals is one of the key properties of a bio-inspired swarming behavior. In the case of swarms of drones, this should result in the ability to properly manage some unattended drones unavailability or critical system faults. We aim at evaluating the swarms robustness in the future works by simulating such occurrences.

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

Agrawal, A. P. and Kaur, A. (2018). A comprehensive comparison of ant colony and hybrid particle swarm optimization algorithms through test case selection. In *Data Engineering and Intelligent Computing*, pages 397–405. Springer.

Alfeo, A. L., Cimino, M. G., De Francesco, N., Lazzeri, A., Lega, M., and Vaglini, G. (2018). Swarm coordination of mini-uavs for target search using imperfect sensors. *Intelligent Decision Technologies*, (Preprint):1–14.

Atten, C., Channouf, L., Danoy, G., and Bouvry, P. (2016). Uav fleet mobility model with multiple pheromones for tracking moving observation targets. In *European Conference on the Applications of Evolutionary Computation*, pages 332–347. Springer.

Aznar, F., Sempere, M., Pujol, M., Rizo, R., and Pujol, M. (2014). Modelling oil-spill detection with swarm drones. In *Abstract and Applied Analysis*, volume 2014. Hindawi.

Bayındır, L. (2016). A review of swarm robotics tasks. *Neurocomputing*, 172:292–321.

Birdsey, L., Szabo, C., and Falkner, K. (2017). Identifying self-organization and adaptability in complex adaptive systems. In *Self-Adaptive and Self-Organizing Systems (SASO), 2017 IEEE 11th International Conference on*, pages 131–140. IEEE.

Bloembergen, D., Tuyls, K., Hennes, D., and Kaisers, M. (2015). Evolutionary dynamics of multi-agent learning: a survey. *Journal of Artificial Intelligence Research*, 53:659–697.

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

Brust, M. R., Zurad, M., Hentges, L., Gomes, L., Danoy, G., and Bouvry, P. (2017). Target tracking optimization of uav swarms based on dual-pheromone clustering. In *Cybernetics (CYBCONF), 2017 3rd IEEE International Conference on*, pages 1–8. IEEE.

Cabreira, T., Brisolara, L., and R Ferreira, P. (2019). Survey on coverage path planning with unmanned aerial vehicles. *Drones*, 3(1):4.

Chiaraviglio, N., Artés, T., Bocca, R., López, J., Gentile, A., Ayanz, J. S. M., Cortés, A., and Margalef, T. (2016). Automatic fire perimeter determination using modis hotspots information. In *e-Science (e-Science), 2016 IEEE 12th International Conference on*, pages 414–423. IEEE.

Cimino, M. G., Lazzeri, A., and Vaglini, G. (2015a). 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*, pages 1–6. IEEE.

Cimino, M. G., Lazzeri, A., and Vaglini, G. (2015b). Improving the analysis of context-aware information via marker-based stigmergy and differential evolution. In *International Conference on Artificial Intelligence and Soft Computing*, pages 341–352. Springer.

Cimino, M. G., Lazzeri, A., and Vaglini, G. (2016). Using differential evolution to improve pheromone-based coordination of swarms of drones for collaborative target detection. In *ICPRAM*, pages 605–610.

Cimino, M. G., Monaco, M., Piscione, P., and Alfeo, A. L. (2018). Github platform, sciadro 3.1 repository. https://github.com/mlpi-unipi/drones-swarm. [Online; Accessed November 10, 2018].

Colby, M., Chung, J. J., and Tumer, K. (2015). Implicit adaptive multi-robot coordination in dynamic environments. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 5168–5173. IEEE.

Cruz, H., Eckert, M., Meneses, J., and Martínez, J.-F. (2016). Efficient forest fire detection index for application in unmanned aerial systems (uass). *Sensors*, 16(6):893.

Das, S., Mullick, S. S., and Suganthan, P. N. (2016). Recent advances in differential evolution–an updated survey. *Swarm and Evolutionary Computation*, 27:1–30.

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

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

DJI (2016). DJIinspire1 specification. https://www.dji.com/inspire-1/info#specs. [Online; accessed May-2018].

DJI (2017). DJIinspire2 specification. https://www.dji.com/inspire-2/info#specs. [Online; accessed May-2018].

Gade, R. and Moeslund, T. B. (2014). Thermal cameras and applications: a survey. *Machine vision and applications*, 25(1):245–262.

Hauert, S., Leven, S., Varga, M., Ruini, F., Cangelosi, A., Zufferey, J.-C., and Floreano, D. (2011). 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*, pages 5015–5020. IEEE.

Hayat, S., Yanmaz, E., and Muzaffar, R. (2016). Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. *IEEE Communications Surveys and Tutorials*, 18(4):2624–2661.

Israel, M. (2011). A uav-based roe deer fawn detection system. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(1/C22):51–55.

Jones, K. O. and Bouffet, A. (2007). Comparison of ant colony optimisation and differential evolution. In *Proceedings of the 2007 international conference on Computer systems and technologies*, page 25. ACM.

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

Kaminka, G. A., Schechter-Glick, R., and Sadov, V. (2008). Using sensor morphology for multirobot formations. *IEEE Transactions on Robotics*, 24(2):271–282.

Kuyucu, T., Tanev, I., and Shimohara, K. (2015). Superadditive effect of multi-robot coordination in the exploration of unknown environments via stigmergy. *Neurocomputing*, 148:83–90.

Labella, T. H., Dorigo, M., and Deneubourg, J.-L. (2006). Division of labor in a group of robots inspired by ants' foraging behavior. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 1(1):4–25.

Loh, W.-K. and Park, Y.-H. (2014). A survey on density-based clustering algorithms. In *Ubiquitous Information Technologies and Applications*, pages 775–780. Springer.

Mathworks (2018). Matlab2018. http://www.mathworks.com. [Online accessed Dec-2018].

Maza, I., Ollero, A., Casado, E., and Scarlatti, D. (2015). Classification of multi-uav architectures. In *Handbook of unmanned aerial vehicles*, pages 953–975. Springer.

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

Meng, W., He, Z., Su, R., Shehabinia, A. R., Lin, L., Teo, R., and Xie, L. (2014). Decentralized control of multi-uavs for target search, tasking and tracking. *IFAC Proceedings Volumes*, 47(3):10048–10053.

Netlogo (2017). http://ccl.northwestern.edu/netlogo/. [Online; accessed Sep-2017].

Paradzik, M. and İnce, G. (2016). Multi-agent search strategy based on digital pheromones for uavs. In *Signal Processing and Communication Application Conference (SIU), 2016 24th*, pages 233–236. IEEE.

Parunak, H. V., Purcell, M., and O'Connell, R. (2002). Digital pheromones for autonomous coordination of swarming uav's. In *1st UAV Conference*, page 3446.

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

Quintero, S. A., Collins, G. E., and Hespanha, J. P. (2013). Flocking with fixed-wing uavs for distributed sensing: A stochastic optimal control approach. In *American Control Conference (ACC), 2013*, pages 2025–2031. IEEE.

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

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH computer graphics*, volume 21, pages 25–34. ACM.

Rodriguez, J., Castiblanco, C., Mondragon, I., and Colorado, J. (2014). Low-cost quadrotor applied for visual detection of landmine-like objects. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 83–88. IEEE.

Sauter, J. A., Matthews, R., Van Dyke Parunak, H., and Brueckner, S. A. (2005a). Performance of digital pheromones for swarming vehicle control. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 903–910. ACM.

Sauter, J. A., Matthews, R., Van Dyke Parunak, H., and Brueckner, S. A. (2005b). Performance of digital pheromones for swarming vehicle control. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 903–910. ACM.

Schmickl, T., Thenius, R., Moslinger, C., Timmis, J., Tyrrell, A., Read, M., Hilder, J., Halloy, J., Campo, A., Stefanini, C., et al. (2011). Cocoro–the self-aware underwater swarm. In *Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2011 Fifth IEEE Conference on*, pages 120–126. IEEE.

See-demining (2018). http://www.seedemining.org/main.htm. [Online; accessed Feb-2018].

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

Singh, S., Lu, S., Kokar, M. M., Kogut, P. A., and Martin, L. (2017). Detection and classification of emergent behaviors using multi-agent simulation framework (wip). In *Proceedings of the Symposium on Modeling and Simulation of Complexity in Intelligent, Adaptive and Autonomous Systems*, page 3. Society for Computer Simulation International.

Suárez Fernández-Miranda, A., Heredia Benot, G., and Ollero Baturone, A. (2016). Cooperative sensor fault recovery in multi-uav systems. In *IEEE International Conference on Robotics and Automation (ICRA),(2016), p 1188-1193*, pages 1188–1193. Elsevier.

Trashout (2018). https://www.trashout.ngo/trashmap/?ref=trashout.me. [Online; accessed Feb-2018].

Vásárhelyi, G., Virágh, C., Somorjai, G., Tarcai, N., Szörényi, T., Nepusz, T., and Vicsek, T. (2014). Outdoor flocking and formation flight with autonomous aerial robots. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3866–3873. IEEE.

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