drones

MDPI

Article

SuSy-EnGaD: Surveillance System Enhanced by Games of Drones

Daniel H. Stolfi [1,*], Matthias R. Brust [1], Grégoire Danoy [1,2] and Pascal Bouvry [1,2]

[1] SnT, University of Luxembourg, 6, Avenue de la Fonte, L-4364 Luxembourg, Luxembourg; matthias.brust@uni.lu (M.R.B.); gregoire.danoy@uni.lu (G.D.); pascal.bouvry@uni.lu (P.B.)
[2] FSTM/DCS, University of Luxembourg, 6, Avenue de la Fonte, L-4364 Luxembourg, Luxembourg
* Correspondence: daniel.stolfi@uni.lu

Abstract: In this article, we propose SuSy-EnGaD, a surveillance system enhanced by games of drones. We propose three different approaches to optimise a swarm of UAVs for improving intruder detection, two of them featuring a multi-objective optimisation approach, while the third approach relates to the evolutionary game theory where three different strategies based on games are proposed. We test our system on four different case studies, analyse the results presented as Pareto fronts in terms of flying time and area coverage, and compare them with the single-objective optimisation results from games. Finally, an analysis of the UAVs trajectories is performed to help understand the results achieved.

Keywords: swarm robotics; unmanned aerial vehicle; evolutionary game theory; evolutionary algorithm; surveillance system; multi-objective optimisation

check for updates

Citation: Stolfi, D.H.; Brust, M.R.; Danoy, G.; Bouvry, P. SuSy-EnGaD: Surveillance System Enhanced by Games of Drones. *Drones* **2022**, *6*, 13. https://doi.org/10.3390/drones6010013

Academic Editors: Diego González-Aguilera and Pablo Rodríguez-Gonzálvez

Received: 1 December 2021
Accepted: 3 January 2022
Published: 6 January 2022

1. Introduction

Unmanned aerial vehicles (UAVs) have a wide range of applications [1] such as wildfire monitoring [2], smart farming [3], aerial surveillance [4], environmental monitoring [5], goods transportation [6] and structural damage mapping [7]. Particularly, surveillance systems [8] require resilience and flexibility to address issues such as drone failures and adverse communications. They can be achieved using swarm intelligence [9] as a way of modifying the collective behaviour of the swarm by using individual parameters. The members' interactions usually follow local rules based on pheromones and probabilities [10], upper confidence trees [11] or finite games [12], among others [13]. These strategies can be based on competitions [14] or collaborations [15] between members, having each approach its own advantages and disadvantages [16]. The associated high dimensional search space makes these problems good candidates to be solved using an intelligent bio-inspired technique, such as evolutionary algorithms [17].

Game-based approaches are commonly present in UAV swarms as a strategy of cooperation between members. Games have been used for cooperative search and surveillance [18], energy-efficient communications [19], beyond-visual-range air combat missions [20], Vehicular Ad hoc NETworks (VANET) communications under adverse network conditions [21], etc. In Evolutionary Game Theory (EGT) [22], group interactions are modelled with the assumption that the surviving strategy is the one which reports outcomes higher than other possible strategies.

In this research work, we propose a surveillance system based on a swarm of UAVs patrolling an area divided into concentric security rings. UAVs in inner rings move slower than those in outer rings but consume also less energy, which allows them to fly for a longer period of time. This surveillance scheme features a compromise between maximum flying time (most of UAVs in the innermost ring) and maximum area coverage (a smart strategy to fly by different rings according to each UAV's battery state). Consequently, our contributions can be summarised as follows:

1. A new Surveillance System Enhanced by Games of Drones (SuSy-EnGaD), based on cooperative UAVs that explore the area of interest arranged in concentric rings.
2. Three different approaches to obtain the optimal strategy taking into account maximum flying time and area coverage.
3. Two bio-inspired evolutionary algorithms adapted to this problem based on the well-known NSGA-II and a genetic algorithm.

The remainder of this paper is organised as follows. In the next section, we review the state of the art related to our work. In Section 3, the SuSy-EnGaD architecture is presented. Our optimisation algorithms are explained in Section 4 and the case studies are described in Section 5. The experimentation done using simulations is presented in Section 6, as well as the discussion of results. After that, Section 7 brings conclusions and future work.

2. Related Work

In this section, we review research works related to our proposal. First, we go through the multi-objective optimisation of UAV related problems. Second, energy optimisation proposals are analysed. Third, surveillance systems using UAVs are reviewed. And finally, cooperative and competitive strategies for UAV swarms are commented.

Many problems related to UAVs are often modelled and solved as multi-objective ones. In [23], a multi-objective path planning framework is proposed to explore a suitable path for a UAV operating in a dynamic urban environment. The authors use safety index maps to capture static (offline search) and dynamic (online search) obstacles. This path planning problem is addressed taking into account two objectives: shorten the travel time and avoid obstacles. A multi-objective optimisation algorithm to allocate tasks and plan paths for a team of UAVs is presented in [24]. A genetic algorithm is used to minimise the mission completion time and can be tuned to prioritise coverage or connectivity. Results obtained via simulations indicate that by transmitting lower rate notifications in the network, the mission time can be shortened. In [25] a multi-objective path planning approach based on the Crowd Distance-based NSGA-II (CDNSGA-II) method is proposed to find an optimal collision-free path for UAVs, taking into account both distance and safety. Experimental results show that the proposed algorithm can obtain up to 80% pareto optimal solutions (see definitions 1 and 2 in Section 3) when compared with NSGA-II under simulated urban environments.

Battery saving is an important concern when using UAVs. An offline path planning algorithm is proposed in [26] to ensure that UAVs have permanent connectivity and can always reach the base station to recharge their batteries. Different approaches for heterogeneous UAVs and multi-base stations were analysed to obtain safe paths using simulations. In [27], an energy-efficient algorithm to optimise fixed-wing UAVs trajectories is proposed. The authors derive a theoretical model of the propulsion energy consumption of fixed-wing UAVs to define the efficiency of communications. They conclude that an optimised simple circular trajectory maximises the energy efficiency and apply their findings to more general itineraries. In our study, quad-rotor UAVs move inside circular rings using unpredictable zigzag trajectories and can change the flying ring during their mission. As we are proposing a surveillance system, area coverage is taken into account in the optimisation process and our results are obtained using a simulator. In [28], the authors present a novel framework for stochastic UAV-assisted surveillance which considers battery constraints. The system uses energy-efficient random walks for flying patterns and probabilistic inspections. A centralised algorithm based on iterative geometric programming approximation was used to solve the problem and the experimentation was conducted using simulations. We propose random walks for sampling the problem's solution landscape to confirm that it is a multi-objective problem. Our UAVs' trajectories always depend on surveillance rings, random bounce angles, and the possible collaborations between drones.

Surveillance systems using UAVs is one of the most discussed applications of these "eyes in the sky". Coordination between UAVs using chaotic mobility combined with pheromone trails is proposed in [29]. In this approach, vehicles explore the surveillance area using detection cells arranged into concentric square rings, stochastically avoiding high concentrations of pheromones. An Evolutionary Algorithm (EA) optimises pheromone amounts and assigns rings to maximise early intruder detection and protect the base in the centre of the analysed scenarios. The advantages of using surveillance rings are discussed in [29], where that approach is also compared with other strategies. In our present study, we are using actual circular rings and UAVs dynamically change rings during the mission by collaborating each other using different strategies. Our proposed trajectories are not using pheromones and they were evaluated using a simulator that includes the full UAV dynamics. Finally, a multi-objective optimisation and game approach were implemented for our current study.

Path planning and coordination of multiple UAVs to provide convoy protection to ground vehicles is proposed in [30]. Different scenarios with stationary and moving convoys were analysed using UAVs, modelled as Dubins vehicles flying at constant altitude. The authors proposed a coordination strategy and optimal paths, also calculating the minimal number of UAVs required. In [31], an optimal navigation algorithm allowing UAVs to determine their movement locally with a minor use of a central station is presented. UAVs perform surveillance task for a group of moving targets while avoiding obstacles. Simulations results using case studies having up to 600 targets are presented to confirm the system's performance. Heterogeneous multiswarm approaches comprise the use of patrolling UAVs and UGVs (Unmanned Ground Vehicles) to serve as mobile refuelling stations [32]. Emerging UAV-UGV inter-swarm collaborations are analysed in [15], and in [33], UAVs, UGVs and UMVs (Unmanned Maritime Vehicles) are combined to improve detections and area coverage on the land and sea.

Finally, we analyse some research works using Evolutionary Game Theory (EGT) [22] approaches to model problems as well as to discover evolving strategies for optimisation [34]. In EGT, players are populations of individuals who follow mixed strategies when playing bimatrix games. In contrast to classical game theory, the surviving strategy after a group of interactions achieves higher benefits than the case of players making rational choices. In [35], packet forwarding strategies are optimised in a mobile wireless ad hoc network (MANET). A Prisoner's Dilemma (PD) model (see Section 3.3) and an EA are proposed to enforce cooperation between nodes. In [16], competition or cooperation are both analysed as possible strategies for drones mapping a disaster area. The problem is modelled as allocation tasks to robots in a swarm under limited communications and partial information, and is solved by a competitive algorithm and a cooperative one. The latter was reported to allocate more tasks in all the analysed scenarios. A modified binary log-linear learning (BLLL) algorithm is proposed in [36] to solve the covering problem using multiple UAVs. The cooperative search problem is modelled as a potential game and a novel action selection strategy for UAVs is proposed. Experiments simulating different mission environments were designed to evaluate the effectiveness and feasibility of the proposed learning algorithm.

Our research work has parts in common with these articles, e.g., cooperation using EGT, results obtained via simulations and bio-inspired multi-objective optimisation of UAVs' trajectories. However, we propose a surveillance area arranged in concentric circular rings, having each one its own flying constraints, plus three approaches using multi-objective and single-objective evolutionary algorithms to calculate the configurations of the UAVs. Some analysed articles involving UAVs focus on visiting predefined targets frequently. Our surveillance proposal in turn, uses unpredictable trajectories to explore the whole area, balancing coverage and battery consumption, which is something critical for every system using UAVs. To the best of our knowledge, no previous work has proposed the evaluation and use of these five evolving strategies for UAV cooperation in a surveillance system using concentric circular rings and trajectories that cannot be easily predicted by trespassers.

3. Surveillance System Enhanced by Games of Drones (SuSy-EnGaD)

The proposed surveillance architecture is based on M security rings where UAVs patrol the area of interest protecting a central base as illustrated in Figure 1. The UAVs are equipped with a video camera directed downwards to scan the area and detect possible intruders. The altitude of the UAVs is fixed, so that the area scanned is about 10 × 10 m (assumed to be square instead of a 16:9 rectangle for simplicity). Since we are not analysing intruder detection rates in our study, we have calculated the area explored using the current position of each UAV, instead of using a specific camera model. This simplifies the simulation model increasing the overall efficiency without losing accuracy. The innermost ring, R_0, is a no-fly zone from where the UAVs depart and eventually return to recharge their batteries. Initially, all the UAVs are assigned to R_1 and they can move to another ring $R_i \leq R_M, i > 0$ depending on their battery level and a series of autonomous decisions (strategy). Since each ring has its own speed restrictions (V_R), such as $V_{R_1} < V_{R_2} < V_{R_M}$, UAVs in inner rings consume less battery than the others in outer rings (and explore a smaller area). When a UAV reaches the external border it bounces back to the surveillance area with an angle θ_b, randomly calculated such as $-\frac{\pi}{6} \leq \theta_b \leq \frac{\pi}{6}$. On the contrary, when a UAV is on the border of a ring, the bouncing angle is θ_r, randomly calculated in the range $-\frac{\pi}{3} \leq \theta_r \leq \frac{\pi}{3}$. The use of a smaller range for θ_b (compared with θ_r) is to prevent UAVs to patrol the surface out of the surveillance area (secant trajectories) while the use of random values adds unpredictability to the UAVs' trajectories, which is desired in surveillance missions.

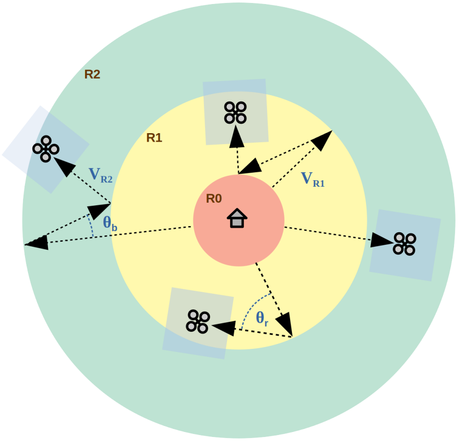
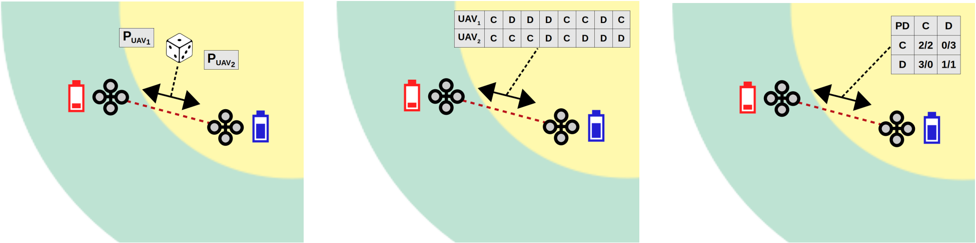


Figure 1. Schema of the Surveillance System Enhanced by Games of Drones (SuSy-EnGaD) where rings, velocities and angles for UAVs are depicted.

The surveillance system as an optimisation problem presents two metrics to be maximised in our study: the area coverage and the vehicles' flying time. By staying in inner rings, UAVs save battery and extend their flying time, but the area coverage is small. On the contrary, if there are too many UAVs in outer rings flying at higher speeds, their batteries will be drained too soon, reducing the system's efficiency.

To dynamically control how UAVs spread by the surveillance area, we propose three approaches for UAVs to decide if they wish to cooperate with their partners by changing rings, or defects and stay in the current ones (Figure 2). Indeed, we call cooperation when a UAV moves to a different flying ring after interacting with another UAV, and defection when it decides to ignore the proposal and stay in its current ring. There exist also a coordination between UAVs when they are in a collision trajectory which is mandatory to keep a safe distance between them (see Section 5). Interactions occur when two UAVs are in their respective communication range (limited to 20 m in our study) where a new ring, in case of cooperation, is calculated according to the relative battery state and the UAV position in the map. Consequently, depending on the decision rules, if both UAVs are in the same ring, the UAV with lower battery charge will move to the next inner ring (if any) and the other will move to the next outer ring (if it cooperates). Otherwise, they will change rings independently of the battery state, which enforces the use of strategies to improve the global metrics of the system by making local decisions (mini-games) that sometimes might be adverse for one or both players (altruism). Despite that, if one UAV (or both) decides not to cooperate, it (they) will stay in the same ring, regardless their battery charge. For efficiency reasons, there is a minimum time between games, i.e., 20 s, and a minimum amount of battery is also required, i.e., 20%. When one UAV has begun a game with another, any other game proposal is also discarded during the next 20 s.



UAV_1	C	D	D	D	C	C	D	C
UAV_2	C	C	C	D	C	D	D	D

PD	C	D
C	2/2	0/3
D	3/0	1/1

(**a**) Probability (**b**) Strategy (**c**) Game

Figure 2. The three proposed approaches for UAVs to decide whether to cooperate and change ring or to defect and stay in the current ring.

The first two approaches to optimising SuSy-EnGaD consist in modelling and solving a multi-objective optimisation problem (MOP). In the *Probability* and *Strategy* approaches, instead of a single solution, a set of pareto optimal solutions are obtained for the two metrics to be maximised: flying time and area coverage. Flying time corresponds to the time at which one of the UAVs in the swarm reaches 10% of battery (giving it enough time for a safe return to base for recharging). Area coverage was measured as the percentage of the area scanned by the onboard cameras (an area of 10 × 10 m).

The main goal of solving a MOP is to find a set of feasible solutions that maximises (in our case, although it can be also defined in terms of minimisation) the objective function vector $\vec{f} = (f_1, \ldots, f_D)$, where D is the number of objectives (two in our study), according to the notion of Pareto-optimality, defined as follows:

Definition 1. *Given the vectors $\vec{x_1}$ and $\vec{x_2}$, we define the solution dominance in a maximisation problem as*

$$\vec{x_1} \prec \vec{x_2} \; if \; \forall i \in \{1, \ldots, D\} f_i(\vec{x_1}) \geq f_i(\vec{x_2}) \tag{1}$$

Definition 2. *The set of solutions not dominated by any other in the solution space S is called the Pareto optimal set (P), defined as*

$$P = \{\vec{x} \in S | \nexists \vec{x\prime} \in S : \vec{x\prime} \prec \vec{x}\} \tag{2}$$

The third approach, *Game*, consists in maximising the average score (outcome) achieved by the UAV swarm as a method for obtaining a solution to the surveillance problem. In this case, it is clearly a single-objective optimisation problem. In the following sections, we describe each approach in detail.

3.1. Probability Approach

In the *Probability* approach each UAV i has a real parameter (p_i) in the range $0.0 \leq p_i \leq 1.0$ defining its cooperation probability. The problem representation is the vector $\vec{x_p}$ composed of N probability values (Equation (3)), where N is the number of UAVs. Different cooperation probability values modify the way UAVs interact, allowing strategies that save battery as well as others that cover a bigger area.

$$\vec{x_p} = \{p_1, p_2, p_3, \ldots, p_N\} \tag{3}$$

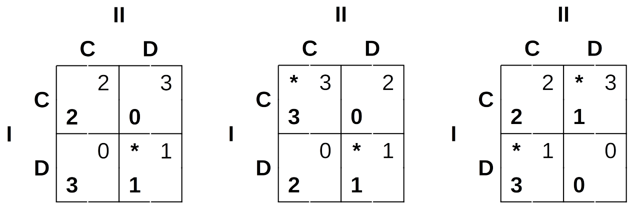
3.2. *Strategy Approach*

The *Strategy* approach proposes for each UAV a strategy defined as a list of bits where "1" means cooperation and "0", defection. The problem representation is the vector $\vec{x_s}$ containing the list of B strategy bits (b_i) for each one of the N UAVs (Equation (4)). In our study, we have set $B = 6$ as a balance between problem complexity (similar to the *Probability* approach) and strategy accuracy. This is a cyclic approach, different from *Probability*, where after six games the strategy of one UAV starts over.

$$\vec{x_s} = \{b_{1_1}, \ldots, b_{B_1}, \ldots, b_{1_N}, \ldots, b_{B_N}\} \tag{4}$$

3.3. *Game Approach*

We propose a third approach based on two-person games. In this case, when two UAVs have to decide a possible cooperation, they will do so by trying to maximise their outcome according to the reward matrices shown in Figure 3, where the Nash equilibria [37] are marked with asterisks.



		II	
		C	D
I	C	2 **2**	3 **0**
	D	0 **3**	* 1 **1**

(**a**) PD

		II	
		C	D
I	C	* 3 **3**	2 **0**
	D	0 **2**	* 1 **1**

(**b**) AG

		II	
		C	D
I	C	2 **2**	* 3 **1**
	D	* 1 **3**	0 **0**

(**c**) CG

Figure 3. Three two-person games: prisoner's dilemma (PD), assurance game (AG), and game of chicken (CG) as proposed in [38]. I and II are the respective players and, C and D stand for cooperation and defection, respectively. Nash equilibria are marked with asterisks.

In the prisoners' dilemma (PD) [38], two prisoners are given the choice of testifying against each other or keeping silent. The best possible outcome is defecting when the other player cooperates (DC). Then comes mutual cooperation (CC) and finally mutual defection (DD) as represented in Figure 3a. The assurance game (AG) (Figure 3b) represents the situation in which a person would be willing to cooperate as long as he/she is assured that their partner would cooperate as well. The better outcome is obtained when there is mutual cooperation (CC), although mutual defection (DD) is also one of the game's equilibria as one person will defect if it thinks that the other will also defect [38]. And finally, the game of chicken (CG), also known as the game of dare, in which two cars are driven one towards the other. The driver who turns away is "chicken" and loses while the other wins. Of course, if no one turns, both drivers die and lose (DD) [38]. There are also two equilibria in CG, unilateral cooperation (CD) and unilateral defection (DC) as shown in Figure 3c.

In the *Game* approach, the UAVs are engaged in a finite two-person game with limited interaction [39]. As the number of iterations is given by how frequently the UAVs meet each other during the simulation, we propose to calculate the best strategy for each player using a bio-inspired meta-heuristic: a genetic algorithm. The *Game* approach uses the same problem representation as in the *Probability* approach (Equation (3)), although, in this case, the objective is maximising the average score of UAVs by modifying their cooperation probability. The evaluation function for this single-objective problem is shown in Equation (5). There, *outcome* refers to the average score achieved by a given UAV after a number of iterations (games) with the other UAVs, obtained from the simulation.

$$F(\vec{x}) = \frac{1}{N} \sum_{i}^{N} outcome(UAV_i) \tag{5}$$

4. Optimisation Algorithms

Multi-objective optimisation problems are usually solved by using one of the following techniques: (i) converting the multi-objective problem into a single-objective one by using a weighted-sum function and (ii) getting a set of Pareto optimal solutions and selecting the most appropriated according to an expert's criterium. The former, despite its simplicity, implies a difficult weight selection that restricts the working solutions during the optimisation process. On the other hand, the latter involves the use of a more complex algorithm, although it increases the versatility and does not require the normalisation of the objectives into the same order of magnitude.

We propose two evolutionary bio-inspired techniques to optimise the parameters of our three approaches. These are efficient methods for solving combinatorial optimisation problems which simulate processes present in evolution such as natural selection, gene recombination after reproduction, gene mutation, and the dominance of the fittest individuals over the weaker ones. Our proposed optimisation algorithms are the Non-dominated Sorting Genetic Algorithm (NSGA-II) for the *Probability* and *Strategy* approaches (multi-objective) and a generational Genetic Algorithm (GA) for the *Game* approach (single-objective).

These optimisation algorithms are to be run in a computer cluster to obtain the optimal solutions for each case study in an initial configuration stage (offline). The solutions achieved, consisting of the optimal configuration of the UAV swarm for each case study, are to be used as the parameters for each UAV to rule the possible collaborations during the surveillance missions (online).

4.1. Non-Dominated Sorting Genetic Algorithm (NSGA-II)

NSGA-II [40] is a multi-objective optimisation algorithm that implements a non-dominated sorting approach to rank solutions based on their Pareto dominance relation. Following the pseudocode in Algorithm 1, after initialising the population composed by $N_i = 28$ individuals, the main loop is executed while the termination condition holds (3000 evaluations in our case). The operators used were chosen according to our problem characteristic and representation. The selection operator is Binary Tournament, Single Point Crossover ($P_c = 0.9$), Integer Polynomial Mutation for the *Probability* approach and Bit Flip Mutation for the *Strategy* approach (for both $P_m = \frac{1}{L}$, where L is the length of the configuration vector), and finally, the replacement operator uses the Ranking and Crowding distance selection for maintaining solution diversity [40].

Algorithm 1. Pseudocode of NSGA-II.

```
procedure NSGA-II($N_i, P_c, P_m$)
    $t \leftarrow 0$
    $Q(0) \leftarrow \emptyset$                                   ▷ Q=auxiliary population
    $P(0) \leftarrow Initialisation(N_i)$                          ▷ P=population
    while not $Termination\_Condition()$ do
        $Q(t) \leftarrow Selection(P(t))$
        $Q(t) \leftarrow Crossover(Q(t), P_c)$
        $Q(t) \leftarrow Mutation(Q(t), P_m)$
        $Evaluation(Q(t))$
        $R \leftarrow Ranking\_and\_Crowding(Q(t), P(t))$
        $P(t+1) \leftarrow Select\_Best\_Individuals(R)$
        $t \leftarrow t + 1$
    end while
end procedure
```

4.2. Genetic Algorithm (GA)

The proposed GA [41,42] is a single-objective optimisation algorithm featuring a population of individuals which evolve to maximise (in our case) their fitness value. The pseudocode of GA is presented in Algorithm 2. Initially, 28 individuals (μ) for the population P are generated and while the termination condition holds the main loop is executed. This is a generational GA where the working population Q has the same number of individuals as P ($\lambda = \mu$). We used Binary Tournament as selection operator, Single Point Crossover as recombination operator, Integer Polynomial Mutation as mutation operator, and an elitist replacement. The parameters of GA are the same as in NSGA-II, i.e., 3000 evaluations, $Pc = 0.9$ and $Pm = \frac{1}{L}$.

Algorithm 2. Pseudocode of GA.

```
procedure GA($N_i, P_c, P_m$)
    $t \leftarrow 0$
    $Q(0) \leftarrow \emptyset$                                  ▷ Q=auxiliary population
    $P(0) \leftarrow Initialisation(N_i)$                         ▷ P=population
    while not $Termination\_Condition()$ do
        $Q(t) \leftarrow Selection(P(t))$
        $Q(t) \leftarrow Crossover(Q(t), P_c)$
        $Q(t) \leftarrow Mutation(Q(t), P_m)$
        $Evaluation(Q(t))$
        $P(t+1) \leftarrow Replacement(Q(t), P(t))$
        $t \leftarrow t+1$
    end while
end procedure
```

4.3. Genetic Operators

In this section, we describe the operators used in both algorithms. We have used the implementation provided by the jMetalPy package [43] for the operators and both optimisation algorithms.

4.3.1. Selection

We have used Binary Tournament [44] as selection operator. It is described in Algorithm 3 where two random samples are taken from the population Q. If it is the multi-objective case (NSGA-II), each sample comprises a pareto from the population while in the case of single-objective optimisation (GA), each sample is an individual representing one configuration of the surveillance system. Then, the samples are compared in terms of dominance (pareto fronts) or fitness (system configurations) and the best of them is included in Q', which will become the working population for the current generation. This process is repeated until the λ required individuals are obtained.

4.3.2. Crossover

Single Point Crossover [45] was used as crossover operator in both algorithms. As shown in Algorithm 4, two individuals $\vec{x}$ and $\vec{y}$ are taken from the population Q to be recombined subject to the crossover probability $P_c = 0.9$. After selecting the crossing point p, the components of $\vec{x}$ and $\vec{y}$ are swapped from the position p to the end of each solution vector. The new resulting vectors $\vec{x}'$ and $\vec{y}'$ are now added to the new working population Q'. By doing so, the optimisation algorithm explores different areas of the solution space in parallel, searching for new promising configurations for the swarm.

Algorithm 3. Pseudocode of the Binary Tournament Selection Operator.

```
function SELECTION($Q$)
    $Q' \leftarrow \emptyset$
    for $i \leftarrow 1, \lambda$ do                              ▷ $Q'$ will have $\lambda$ individuals
        $s_1 \leftarrow random\_sample(Q)$          ▷ randomly takes two individuals from $Q$
        $s_2 \leftarrow random\_sample(Q)$              ▷ they are pareto fronts in NSGA-II
        if $better(s_1, s_2)$ then              ▷ compare in terms of fitness/dominance
            $Q' \leftarrow Q' \cup \{s_1\}$
        else
            $Q' \leftarrow Q' \cup \{s_2\}$
        end if
    end for
    return $Q'$
end function
```

Algorithm 4. Pseudocode of the Single Point Crossover Operator.

```
function CROSSOVER($Q, P_c$)
    $Q' \leftarrow \emptyset$
    for all $\{\vec{x}, \vec{y}\} \in Q$ do            ▷ all the individuals in $Q$, taken in pairs
        $\vec{x}' = \vec{x}$
        $\vec{y}' = \vec{y}$
        if $rnd() < P_c$ then                                    ▷ crossover probability
            $p \leftarrow randInt(1, L)$   ▷ crossing point $p$, $L$ is the length of the solution vector
            for $i \leftarrow p, L)$ do                ▷ swaps vector's components from $p$ to $L$
                $\vec{x}'[i] = \vec{y}[i]$
                $\vec{y}'[i] = \vec{x}[i]$
            end for
        end if
        $Q' \leftarrow Q' \cup \{\vec{x}', \vec{y}'\}$                ▷ adds the new vectors to the result
    end for
    return $Q'$
end function
```

4.3.3. Mutation

Two mutation operators were used. Bit Flip Mutation for the binary representation (*Strategy* approach) and Integer Polynomial Mutation for the *Probability* and *Game* approaches. They are meant to make small variations to the configuration vectors and explore the neighbourhood of the good solutions already found. The selected value for $P_m = \frac{1}{L}$ stochastically selects one position of each solution vector (individual of the working population) to be changed. The Bit Flip Mutation [46] consists in randomly selecting bits in the solution vector to be flipped as shown in Algorithm 5. After all the individuals were considered for a possible mutation the resulting population Q' is returned.

The Integer Polynomial Mutation [47] was selected to work with the numeric values in the vector of probabilities. To simplify the search process we have worked with integer values between 0 and 100 to represent probabilities between 0.0 and 1.0 with an accuracy of two decimal places. The Algorithm 6 shows the pseudocode of this operator. It can be seen that for each position in the solution vector subject to be mutated, four values are calculated depending on the current position's value, the maximum allowed value (100), and a new random value (ρ) as well. A Δ_q value is calculated depending on ρ, to modify the original value, subject to the right range of values.

Algorithm 5. Pseudocode of the Bit Flip Mutation Operator.

```
function BITFLIPMUTATION($Q, P_m$)
    $Q' \leftarrow \emptyset$
    for all $\{\vec{x}\} \in Q$ do                                    ▷ all the individuals in $Q$
        $\vec{x}' \leftarrow \vec{x}$
        for $i \leftarrow 1, L$ do          ▷ $i$ goes from 1 to $L$ (the length of the solution vector)
            if $rnd() < P_m$ then                                         ▷ mutation probability
                $\vec{x}'[i] \leftarrow not(\vec{x}[i])$                                    ▷ bit flip
            end if
        end for
        $Q' \leftarrow Q' \cup \{\vec{x}'\}$
    end for
    return $Q'$
end function
```

Algorithm 6. Pseudocode of the Integer Polynomial Mutation Operator.

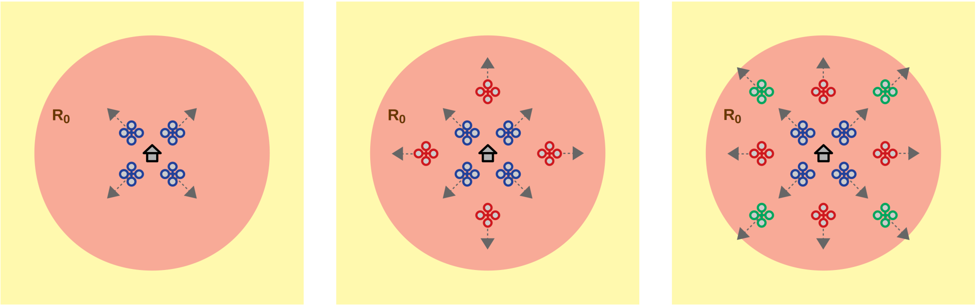
```
function INTEGERPOLYNOMIALMUTATION($Q, P_m$)
    $Q' \leftarrow \emptyset$
    for all $\{\vec{x}\} \in Q$ do                                    ▷ all the individuals in $Q$
        $\vec{x}' \leftarrow \vec{x}$
        for $i \leftarrow 1, L$ do          ▷ $i$ goes from 1 to $L$ (the length of the solution vector)
            if $rnd() < P_m$ then                                         ▷ mutation probability
                $\delta_1 \leftarrow \frac{\vec{x}'[i]}{100}; \delta_2 \leftarrow \frac{100-\vec{x}'[i]}{100}; \pi = \frac{1}{\sigma+1}; \rho \leftarrow rnd()$          ▷ $\sigma = 0.2$ by default
                if $\rho \leq 0.5$ then                                               ▷ equiprobable
                    $\Delta_q = (2 \times \rho + (1 - 2 \times \rho) \times (1 - \delta_1)^{\sigma+1})^{\pi-1}$
                else
                    $\Delta_q = 1 - (2 \times (1 - \rho) + (2 \times (\rho - 0.5) \times (1 - \delta_2)^{\sigma+1})^{\pi}$
                end if
                $y \leftarrow Bounds(\vec{x}'[i] + Delta_q \times 100)_{0,100}$          ▷ keeps it in range [0, 100]
                $\vec{x}'[i] \leftarrow round(y)$                                  ▷ nearest integer value
            end if
        end for
        $Q' \leftarrow Q' \cup \{\vec{x}'\}$
    end for
    return $Q'$
end function
```

5. Case Studies

To evaluate SuSy-EnGaD using the three approaches previously discussed, four case studies are proposed featuring different numbers of UAVs and map dimensions. The characteristics of the case studies are detailed in Table 1, as well as the nomenclature used, i.e., UAVS.MAPSIZE. The innermost ring corresponds to the area from where the UAVs depart, and after this initial stage, it becomes a no-fly zone. The evaluation of the UAVs configuration, according to each approach, was performed using the ARGoS simulator [48]. ARGoS is a multi-physics robot simulator that can simulate large-scale swarms of robots of any kind, efficiently. In our study, we simulate UAVs using the eye-bot drones [49] provided by ARGoS, including their communication and battery consumption models. This allows us to efficiently test multiple configurations keeping in mind reliability and safety since a wrong configuration does not end in a catastrophic collision while many scenarios can be evaluated in parallel. Depending on the number of robots, the departing point and the initial heading angle was predefined to avoid excessive use of the collision avoidance algorithm at the beginning of the simulation. Figure 4 shows the initial configuration for 4, 8, and 12 UAVs, all departing from the innermost ring (in red).

Table 1. Characteristics of the Four Case Studies.

Case Study	UAVs	Map Size (m)	Rings	Radii
4.100	4	100 × 100	3	5/28/50
8.100	8	100 × 100	3	5/28/50
8.200	8	200 × 200	3	5/53/100
12.200	12	200 × 200	3	5/53/100



(**a**) 4 UAVs (**b**) 8 UAVs (**c**) 12 UAVs

Figure 4. Departure points and initial directions for case studies comprising 4, 8 and 12 UAVs.

The designed collision avoidance algorithm relies on repelling forces between UAVs as described in Algorithm 7. Given $u \in UAVs$, the distances between u and the rest of vehicles in $UAVs$ are calculated. Those UAVs closer than a minimum distance δ_{min} (a fixed parameter, e.g., 9 m) modify the vector $\vec{r}$, which will contain the resultant repelling force. As a result of this coordination, $\vec{r}$ is to be used to modify the original trajectory of uav.

Algorithm 7. Collision Avoidance Algorithm.

```
function CollisionAvoidance($uav, UAVs$)
    $\vec{r} \leftarrow (0,0)$
    for all $u, \in UAVs$ do                                  ▷ all the UAVs
        if $u \neq uav$ then
            $\delta \leftarrow \sqrt{(uav_x - u_x)^2 + (uav_y - u_y)^2}$
            if $\delta < \delta_{min}$ then                   ▷ $\delta_{min} = 9$
                $\vec{r} \leftarrow \vec{r} - (u_x, u_y)$
            end if
        end if
    end for
    return $\vec{r}$
end function
```

6. Simulation Results

First, we analyse the proposed metrics, flying time and area coverage, to confirm that the use of a multi-objective optimisation is appropriated. Second, we have optimised the four case studies using NSGA-II for the *Probability* and *Strategy* approaches. Third, we have used the proposed GA to maximise the average score obtained by the UAVs in the four case studies using the *Game* approach (three two-person games). Finally, an analysis of the UAVs' trajectories is given.

We have used the jMetalPy package [43] to implement both optimisation algorithms. Our experiments were performed doing parallel runs in the HPC facilities of the University of Luxembourg [50]. Since the use of the ARGoS simulator implies long evaluation times to preserve the realism (10 simulation ticks per second), we have evaluated each algorithm' generation (28 individuals) in parallel on computing nodes equipped with Intel Xeon Gold

6132 2.6 GHz and 128 GB of RAM. The total optimisation time (600 runs in total) was equivalent to 1750.8 h (about 73 days).

6.1. Metric Analysis

In this experiment, we analyse the existence of correlations between the metrics used in our study. We have performed 30 random walks of 20 steps each, for 4, 8 and 12 UAVs, (we have considered that 1800 steps in total are enough for getting a significant sample of the problem's solutions) and calculated the Pearson correlation coefficients [51] shown in Table 2. Note that the random walks are performed on the system configuration, i.e., a vector of probabilities as defined in Equation (3), to sample the solution landscape of the problem. The UAVs are always following the SuSy-EnGaD trajectories based on surveillance rings.

It can be seen that 4 UAVs present some negative correlation between flying time and area coverage, so that one increases when the other decreases and vice versa. On the contrary, the metrics of 8 and 12 UAVs are not correlated (Pearson coefficient approximately equal to 0). Consequently, we can confirm that the use of a multi-objective optimisation algorithm (NSGA-II) is appropriate to obtain not one but many non-dominated solutions to our surveillance problem. After that, an expert can decide which solution suits better its needs and whether higher coverage levels could be prioritised against flying times or the other way round.

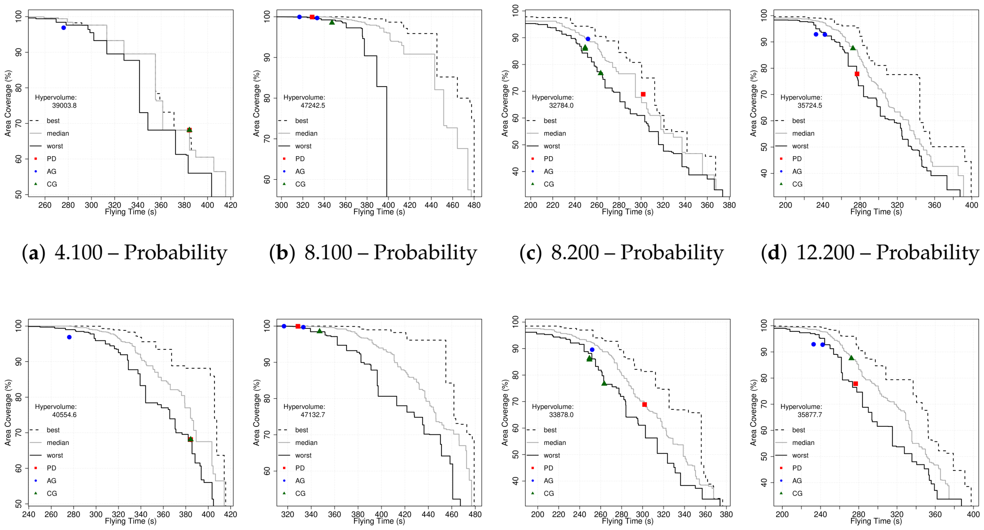
Table 2. Correlation Matrix for 4, 8 and 12 UAVs.

Metric	**4 UAVs**		**8 UAVs**		**12 UAVs**	
	F. Time	**Coverage**	**F. Time**	**Coverage**	**F. Time**	**Coverage**
F. Time	1.000	−0.599	1.000	−0.051	1.000	−0.006
Coverage	−0.599	1.000	−0.051	1.000	−0.006	1.000

6.2. Multi-Objective Optimisation

The next experiment consisted of the optimisation of the four case studies using a multi-objective algorithm (NSGA-II). We have performed 30 runs for the *Probability* and *Strategy* approaches on each case study (240 runs of NSGA-II in total) to get the results shown in Figure 5. We have plotted the empirical attainment functions (EAF) [52] describing the probabilistic distribution of the outcomes obtained by NSGA-II. Additionally, the hypervolume of the union of all sets using the reference point 0,0 is reported to compare the results of both approaches on each case study.

We can see that both strategies obtained similar results and that differences between hypervolumes are always under 4%. One optimisation run for 8.100 using the *Probability* approach seems to have suffered from stagnation, which can be clearly seen through the worst-case depicted in Figure 5b. That is one of the reasons why we performed 30 runs of stochastic algorithms. Now, one result from the set of results in each Pareto front could be selected and the associated configuration (whether it is composed by a set of probabilities or a list of bits) is to be used to set up the surveillance system prioritising coverage, flying times or a compromise solution. A different approach based on games is used in the next set of experiments described in the following section.



(**a**) 4.100 – Probability (**b**) 8.100 – Probability (**c**) 8.200 – Probability (**d**) 12.200 – Probability

(**e**) 4.100 – Strategy (**f**) 8.100 – Strategy (**g**) 8.200 – Strategy (**h**) 12.200 – Strategy

Figure 5. Plots of the empirical attainment functions obtained after optimising our case studies using NSGA-II (*Probability* and *Strategy* approaches). Additionally, the best values from *Game* (PD, AG and CG) obtained by GA are drawn. The respective hypervolume values are also reported.

6.3. Game Results

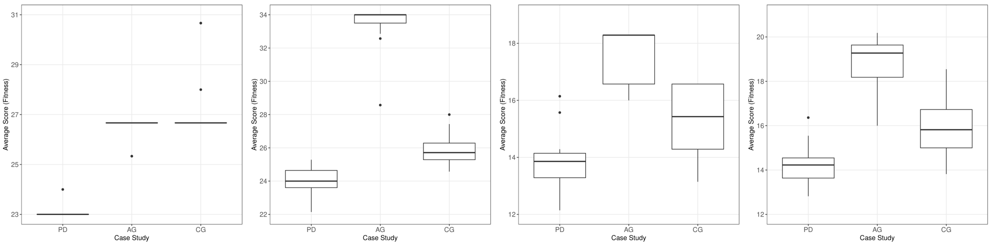
A different optimisation approach consists in using games to model each UAV interaction. As the objective is the maximisation of the average score of the swarm, we have used a single-objective GA. Table 3 shows the result of this optimisation process where 360 runs of GA were performed (30 per game and case study) to perform a reliable statistical analysis since we are working with a stochastic optimisation algorithm. It can be seen that UAVs playing the assurance game (AG) have maximised the average score of the swarm (fitness value). The game of chicken (CG) achieved the best maximum and minimum results for 4.100 although the median is still the same as in AG. All the results are statistically significant as shown by the reported Friedman ranks and Wilcoxon *p*-values.

Table 3. Results of the optimisation of the case studies using GA in the Game approach. Statistical tests are also included. Best values are in bold.

Case Study	Game		Fitness		Friedman	Wilcoxon
		Min.	**Median**	**Max.**	**Rank**	***p*-Value**
4.100	PD	23.0	23.0	24.0	1.00	2.1×10^{-7}
	AG	25.3	**26.7**	26.7	2.40	3.5×10^{-7}
	CG	**26.7**	**26.7**	**30.7**	2.60	—
8.100	PD	22.1	24.0	25.3	1.00	1.8×10^{-6}
	AG	**28.6**	**34.0**	**34.0**	3.00	—
	CG	24.6	24.7	28.0	2.00	1.7×10^{-6}
8.200	PD	12.1	13.9	16.1	1.18	2.0×10^{-6}
	AG	**16.0**	**18.3**	**18.3**	2.90	—
	CG	13.1	15.4	16.6	1.92	2.3×10^{-6}
12.200	PD	12.8	14.2	16.4	1.10	1.8×10^{-6}
	AG	**16.0**	**19.3**	**20.2**	2.93	—
	CG	13.8	15.8	18.5	1.97	3.0×10^{-6}

We have plotted the corresponding metric points (flying time vs. area coverage) over the Pareto fronts graphics in Figure 5. It is interesting to see that AG favours high area coverage values in all case studies, while PD's results are obtaining better flying times in three of the case studies. UAVs playing CG are usually in a middle point between the other two games. Some games present different metric values for the same maximum scores (different trajectories, same game results) as depicted in Figure 5.

As a piece of complementary information about the optimisation process performed by the GA, we show in Figure 6 the boxplots with the distribution of the results achieved for each game and case study. We can see that the scores for AG are higher in most of the cases, except by 4.100 where CG presents some outliers revealing a higher average score (fitness) for the swarm. Looking at the scores in Figure 3, we see that the highest values obtained from cooperation (CC) are provided by AG. Consequently, we can assume that this was the strategy followed in the majority of cases by the UAVs.



(**a**) 4.100 (**b**) 8.100 (**c**) 8.200 (**d**) 12.200

Figure 6. Boxplots showing the distribution of the results from the 30 independent runs of GA for each case study and game.

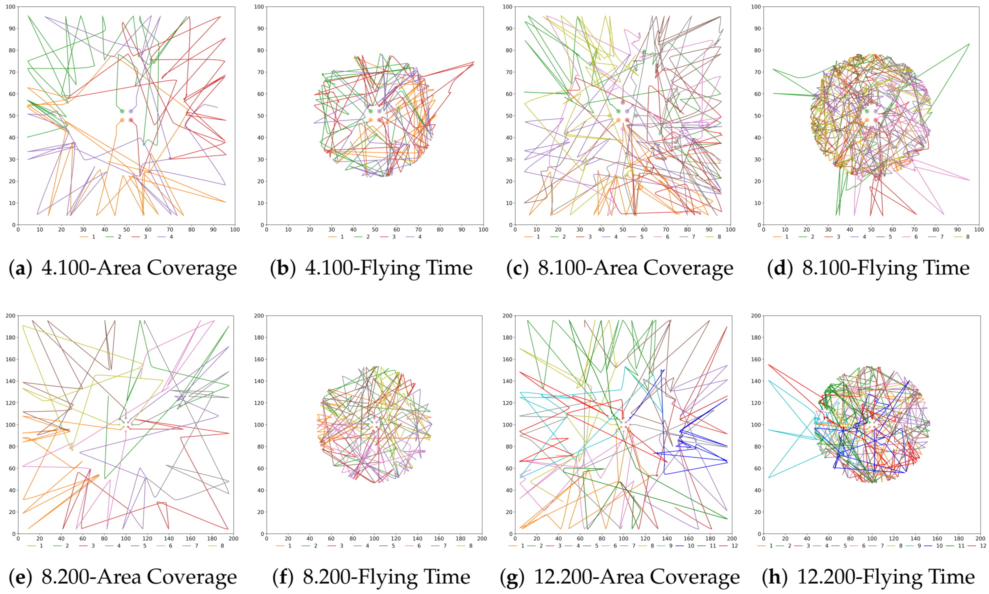
The extreme solutions for all the approaches are shown in Table 4. Note that we have used the default consumption model provided by ARGoS for all our experiments. Hence, maximum flying times ought to be longer than 480 s if other models of drones are used. In the next section, an analysis of the UAVs' trajectories is presented to better understand their behaviour under different configurations.

Table 4. Maximum values obtained after the optimisation of the three approaches. Probability and Strategy values are taken from the Pareto front (NSGA-II) whereas Game values (PD, AG and CG) are the best result obtained by GA. Best values are in bold.

Case Study	**Metric**	**Approach**				
		Probability	**Strategy**	**Game**		
				PD	**AG**	**CG**
4.100	F. Time (s)	**415.6**	**415.6**	384.4	276.1	384.4
	Coverage (%)	**100.0**	**100.0**	68.1	96.9	68.1
8.100	F. Time (s)	**480.1**	479.1	328.7	333.4	347.2
	Coverage (%)	**100.0**	**100.0**	99.9	**100.0**	98.5
8.200	F. Time (s)	373.9	**376.4**	301.9	251.9	263.4
	Coverage (%)	97.8	**98.5**	68.9	89.6	86.3
12.200	F. Time (s)	**399.3**	397.6	276.7	242.2	272.3
	Coverage (%)	99.6	**99.9**	77.8	92.9	87.5

6.4. *Analysis of Games' Trajectories*

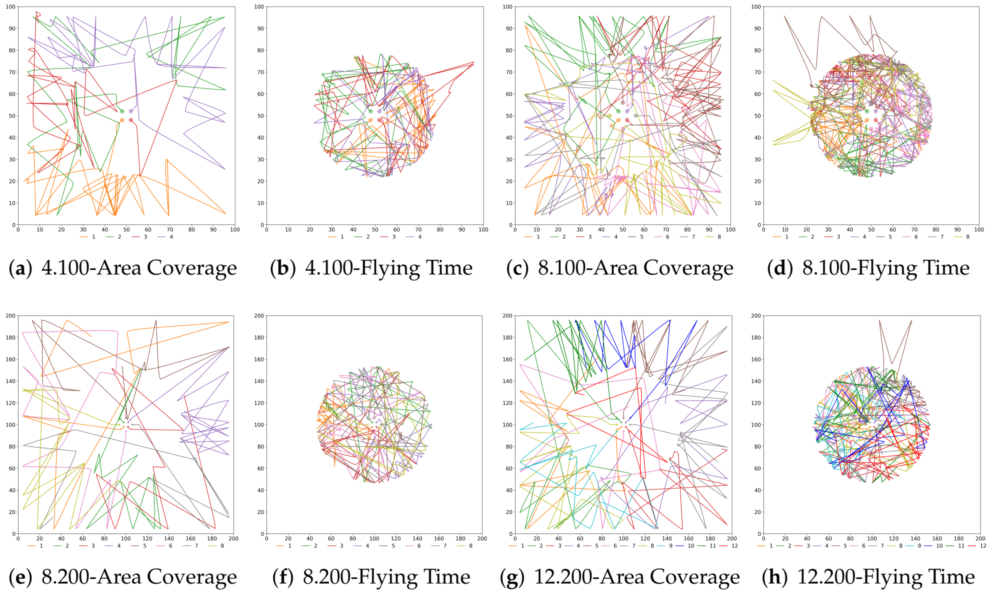
The objective of this last study is to better understand the implications of each configuration and approach and how a cooperation/defection decision modifies the swarm behaviour. We have already presented the numerical results but now in Figures 7 and 8 the UAVs trajectories for the extreme solutions are depicted.



(**a**) 4.100-Area Coverage (**b**) 4.100-Flying Time (**c**) 8.100-Area Coverage (**d**) 8.100-Flying Time

(**e**) 8.200-Area Coverage (**f**) 8.200-Flying Time (**g**) 12.200-Area Coverage (**h**) 12.200-Flying Time

Figure 7. UAVs trajectories for maximum area coverage and maximum flying time using the *Probability* approach.



(**a**) 4.100-Area Coverage (**b**) 4.100-Flying Time (**c**) 8.100-Area Coverage (**d**) 8.100-Flying Time

(**e**) 8.200-Area Coverage (**f**) 8.200-Flying Time (**g**) 12.200-Area Coverage (**h**) 12.200-Flying Time

Figure 8. UAVs trajectories for maximum area coverage and maximum flying time using the *Strategy* approach.

We can see that there exist many similarities in the trajectories when using *Probability* and *Strategy*. In the configurations maximising area coverage, the UAVs depart from the innermost ring and visit the other two rings regularly, where the surveillance mission takes place. On the contrary, when the objective of the swarm is maximising the flying time, most of the UAVs stay in the middle ring after departing from the centre. It is interesting to remark that the maximum flying time was sometimes achieved when some UAVs have a short excursion to the outer ring. That was unexpected as we believed that flying at a lower speed all the time would have been better for saving energy. As we can see, the less congested cases (Figures 7f and 8f) are indeed according to our preliminary assumptions, since the rings' areas are bigger. We believe that multiple iterations in a limited space, plus the effects of the collision avoidance algorithm, have made NSGA-II to find better solutions (in terms of battery consumption) by temporarily sending some UAVs to the outer ring.

7. Conclusions

In this research work, we have presented a new Surveillance System Enhanced by Games of Drones (SuSy-EnGaD), based on cooperative UAVs that explore the surveillance area modelled as concentric rings. We have proposed three different approaches to obtain optimal strategies focusing on maximum flying time and area coverage. The first two approaches, *Probability* and *Strategy* are addressed as multi-objective optimisation problems using the well-known NSGA-II. The third approach is related to the evolutionary game theory where prisoner's dilemma, assurance game and game of chicken are proposed to rule possible cooperation between UAVs. This approach was optimised using a genetic algorithm maximising the average score of the swarm and the results were compared with the multi-objective approaches. We have tested SuSy-EnGaD on four different case studies featuring two map sizes and swarms of four, eight and twelve UAVs. Finally, the UAVs trajectories were analysed to better understand the best configurations in terms of flying time and area coverage.

Our results show that flying times up to 480 s can be achieved with a low area coverage, while on the other hand, a full area coverage is possible when the flying time is roughly under 250 s (depending on the case study). Vehicles' trajectories have indicated a high concentration of UAVs in the inner surveillance ring to save battery (they fly at a reduced speed there), and a more balanced distribution of UAVs throughout the surveillance area to achieve high coverage rates. The single-objective *Game* approach obtained optimal solutions by maximising the swarm's average score. Although the extreme values were obtained by using the multi-objective approaches, the *Game* approach turned out to be a different strategy to obtain particular solutions depending on the game selected.

In our simulations performed using a multi-physics robot simulator (ARGoS), we have observed UAVs' trajectories visiting the entire surveillance scenario when area coverage was prioritised. On the contrary, almost all UAVs stayed in the inner ring when the objective was to keep a maximum flying time. Under this condition, we have observed some UAVs momentarily leaving the inner ring, which we believe is due to the collision avoidance algorithm as several UAVs were flying in a reduced area. These trajectories confirm the expected UAV behaviour and are in accordance with the numeric results obtained.

As a matter of future work we plan to test our proposal using a different number of security rings and UAVs. Previously, we have experimented with square rings and pheromones, in this current article we have improved the surveillance scenario using actual circular rings, and we wish to take advantage of this new architecture in future research works. We would like to try other different games to study the various solutions that can be achieved and also defining our own payoff matrices to obtain different single solution points from the pareto front. Despite the accuracy provided by ARGoS, we are working on increasing the realism of our approach testing the trajectories using actual drones where weather conditions, vision algorithms, and communication restrictions are to be considered. Another interesting future work would consist in using a communication layer featuring different packet loss rates to analyse how it would affect the swarm behaviour, especially the collision avoidance algorithm.

Author Contributions: Conceptualisation, D.H.S.; methodology, D.H.S.; software, D.H.S.; validation, D.H.S. and G.D.; formal analysis, D.H.S.; writing—original draft preparation, D.H.S.; writing—review and editing, D.H.S., M.R.B. and G.D.; visualisation, D.H.S.; supervision, G.D. and P.B.; project administration, G.D. and P.B.; funding acquisition, G.D. and P.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the Office of Naval Research, grant number N62909-18-1-2176.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in https://gitlab.uni.lu/hunted/susy-engad-surveillance-system-enhanced-by-games-of-drones (accessed on 5 January 2021).

Acknowledgments: This work relates to Department of Navy award N62909-18-1-2176 issued by the Office of Naval Research. The United States Government has a royalty-free license throughout the world in all copyrightable material contained herein. This work is partially funded by the joint research programme UL/SnT-ILNAS on Digital Trust for Smart-ICT. The experiments presented in this paper were carried out using the HPC facilities of the University of Luxembourg [50]—see https://hpc.uni.lu (accessed on 5 January 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Skorobogatov, G.; Barrado, C.; Salamí, E. Multiple UAV Systems: A Survey. *Unmanned Syst.* **2020**, *08*, 149–169. [CrossRef]
2. Lin, Z.; Liu, H.H.T.; Wotton, M. Kalman Filter-Based Large-Scale Wildfire Monitoring with a System of UAVs. *IEEE Trans. Ind. Electron.* **2019**, *66*, 606–615. [CrossRef]
3. Lottes, P.; Khanna, R.; Pfeifer, J.; Siegwart, R.; Stachniss, C. UAV-based crop and weed classification for smart farming. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3024–3031. [CrossRef]
4. Bozcan, I.; Kayacan, E. UAV-AdNet: Unsupervised anomaly detection using deep neural networks for aerial surveillance. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October–24 January 2020; pp. 1158–1164. [CrossRef]
5. He, X.; Bourne, J.R.; Steiner, J.A.; Mortensen, C.; Hoffman, K.C.; Dudley, C.J.; Rogers, B.; Cropek, D.M.; Leang, K.K. Autonomous Chemical-Sensing Aerial Robot for Urban/Suburban Environmental Monitoring. *IEEE Syst. J.* **2019**, *13*, 3524–3535. [CrossRef]
6. Yakushiji, K.; Fujita, H.; Murata, M.; Hiroi, N.; Hamabe, Y.; Yakushiji, F. Short-Range Transportation Using Unmanned Aerial Vehicles (UAVs) during Disasters in Japan. *Drones* **2020**, *4*, 68. [CrossRef]
7. Kerle, N.; Nex, F.; Gerke, M.; Duarte, D.; Vetrivel, A. UAV-Based Structural Damage Mapping: A Review. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 14. [CrossRef]
8. Keller, J.; Thakur, D.; Likhachev, M.; Gallier, J.; Kumar, V. Coordinated Path Planning for Fixed-Wing UAS Conducting Persistent Surveillance Missions. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 17–24. [CrossRef]
9. Altan, A. Performance of metaheuristic optimization algorithms based on swarm intelligence in attitude and altitude control of unmanned aerial vehicle for path following. In Proceedings of the 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Istanbul, Turkey, 22–24 October 2020; pp. 1–6. [CrossRef]
10. Yang, F.; Ji, X.; Yang, C.; Li, J.; Li, B. Cooperative search of UAV swarm based on improved ant colony algorithm in uncertain environment. In Proceedings of the 2017 IEEE International Conference on Unmanned Systems (ICUS), Beijing, China, 27–29 October 2017; pp. 231–236. [CrossRef]
11. Wang, T.; Qin, R.; Chen, Y.; Snoussi, H.; Choi, C. A reinforcement learning approach for UAV target searching and tracking. *Multimed. Tools Appl.* **2018**, *78*, 4347–4364. [CrossRef]
12. Weibull, J.W. *Evolutionary Game Theory*; MIT Press: Cambridge, MA, USA, 1997.
13. Zhou, Y.; Rao, B.; Wang, W. UAV Swarm Intelligence: Recent Advances and Future Trends. *IEEE Access* **2020**, *8*, 183856–183878. [CrossRef]
14. Sims, K. Evolving 3D Morphology and Behavior by Competition. *Artif. Life* **1994**, *1*, 353–372. [CrossRef]
15. Stolfi, D.H.; Brust, M.R.; Danoy, G.; Bouvry, P. Emerging Inter-Swarm Collaboration for Surveillance Using Pheromones and Evolutionary Techniques. *Sensors* **2020**, *20*, 2566. [CrossRef] [PubMed]
16. Jesus Roldan, J.; Del Cerro, J.; Barrientos, A. Should we compete or should we cooperate? Applying game theory to task allocation in drone swarms. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 5366–5371. [CrossRef]
17. Bäck, T. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*; Oxford University Press: Oxford, UK, 1996.
18. Li, P.; Duan, H. A potential game approach to multiple UAV cooperative search and surveillance. *Aerosp. Sci. Technol.* **2017**, *68*, 403–415. [CrossRef]
19. Ruan, L.; Wang, J.; Chen, J.; Xu, Y.; Yang, Y.; Jiang, H.; Zhang, Y.; Xu, Y. Energy-efficient multi-UAV coverage deployment in UAV networks: A game-theoretic framework. *China Commun.* **2018**, *15*, 194–209. [CrossRef]
20. Ma, Y.; Wang, G.; Hu, X.; Luo, H.; Lei, X. Cooperative Occupancy Decision Making of Multi-UAV in Beyond-Visual-Range Air Combat: A Game Theory Approach. *IEEE Access* **2020**, *8*, 11624–11634. [CrossRef]
21. Shivshankar, S.; Jamalipour, A. An Evolutionary Game Theory-Based Approach to Cooperation in VANETs under Different Network Conditions. *IEEE Trans. Veh. Technol.* **2015**, *64*, 2015–2022. [CrossRef]

22. Peters, H. *Game Theory*; Springer Texts in Business and Economics; Springer: Berlin/Heidelberg, Germany, 2015; Volume 228, p. 36. [CrossRef]
23. Yin, C.; Xiao, Z.; Cao, X.; Xi, X.; Yang, P.; Wu, D. Offline and Online Search: UAV Multiobjective Path Planning Under Dynamic Urban Environment. *IEEE Internet Things J.* **2018**, *5*, 546–558. [CrossRef]
24. Hayat, S.; Yanmaz, E.; Brown, T.X.; Bettstetter, C. Multi-objective UAV path planning for search and rescue. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5569–5574. [CrossRef]
25. Ren, Q.; Yao, Y.; Yang, G.; Zhou, X. Multi-objective path planning for UAV in the urban environment based on CDNSGA-II. In Proceedings of the 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE), San Francisco, CA, USA, 4–9 April 2019; pp. 350–3505. [CrossRef]
26. Scherer, J.; Rinner, B. Persistent multi-UAV surveillance with energy and communication constraints. In Proceedings of the 2016 IEEE International Conference on Automation Science and Engineering (CASE), Fort Worth, TX, USA, 21–25 August 2016; pp. 1225–1230. [CrossRef]
27. Zeng, Y.; Zhang, R. Energy-Efficient UAV Communication with Trajectory Optimization. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 3747–3760. [CrossRef]
28. Hosseinalipour, S.; Rahmati, A.; Eun, D.Y.; Dai, H. Energy-Aware Stochastic UAV-Assisted Surveillance. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 2820–2837. [CrossRef]
29. Stolfi, D.H.; Brust, M.R.; Danoy, G.; Bouvry, P. CONSOLE: Intruder detection using a UAV swarm and security rings. *Swarm Intell.* **2021**, *15*, 205–235. [CrossRef]
30. Ding, X.C.; Rahmani, A.R.; Egerstedt, M. Multi-UAV Convoy Protection: An Optimal Approach to Path Planning and Coordination. *IEEE Trans. Robot.* **2010**, *26*, 256–268. [CrossRef]
31. Huang, H.; Savkin, A.V. Navigating UAVs for Optimal Monitoring of Groups of Moving Pedestrians or Vehicles. *IEEE Trans. Veh. Technol.* **2021**, *70*, 3891–3896. [CrossRef]
32. Jayavelu, S.; Kandath, H.; Sundaram, S. Dynamic area coverage for multi-UAV using distributed UGVs: A two-stage density estimation approach. In Proceedings of the 2018 Second IEEE International Conference on Robotic Computing (IRC), Laguna Hills, CA, USA, 31 January–2 February 2018; pp. 165–166. [CrossRef]
33. Stolfi, D.H.; Brust, M.R.; Danoy, G.; Bouvry, P. UAV-UGV-UMV Multi-Swarms for Cooperative Surveillance. *Front. Robot. AI* **2021**, *8*, 616950. [CrossRef]
34. Samuelson, L. *Evolutionary Games and Equilibrium Selection*; MIT Press: Cambridge, MA, USA, 1997.
35. Seredynski, M.; Bouvry, P. Evolutionary game theoretical analysis of reputation-based packet forwarding in civilian mobile Ad Hoc networks. In Proceedings of the 2009 IEEE International Symposium on Parallel & Distributed Processing, Rome, Italy, 23–29 May 2009; pp. 1–8. [CrossRef]
36. Ni, J.; Tang, G.; Mo, Z.; Cao, W.; Yang, S.X. An Improved Potential Game Theory Based Method for Multi-UAV Cooperative Search. *IEEE Access* **2020**, *8*, 47787–47796. [CrossRef]
37. Nash, J. Non-cooperative games. *Ann. Math.* **1951**, *54*, 286–295. [CrossRef]
38. Kollock, P. Social Dilemmas: The Anatomy of Cooperation. *Annu. Rev. Sociol.* **1998**, *24*, 183–214. [CrossRef]
39. Seredynski, F. Competitive Coevolutionary Multi-Agent Systems: The Application to Mapping and Scheduling Problems. *J. Parallel Distrib. Comput.* **1997**, *47*, 39–57. [CrossRef]
40. Deb, K.; Agrawal, S.; Pratap, A.; Meyarivan, T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Parallel Problem Solving from Nature PPSN VI*; Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J.J., Schwefel, H.P., Eds.; Springer: Berlin/Heidelberg, Germany, 2000; pp. 849–858.
41. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed.; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 1989.
42. Holland, J.H. *Adaptation in Natural and Artificial Systems*; The MIT Press: Cambridge, MA, USA, 1992; p. 228. [CrossRef]
43. Benítez-Hidalgo, A.; Nebro, A.J.; García-Nieto, J.; Oregi, I.; Ser, J.D. jMetalPy: A Python framework for multi-objective optimization with metaheuristics. *Swarm Evol. Comput.* **2019**, *51*, 100598. [CrossRef]
44. Goldberg, D.E.; Deb, K. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. *Found. Genet. Algorithms* **1991**, *1*, 69–93. [CrossRef]
45. De Jong, K.A. An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Ph.D. Thesis, University of Michigan, Ann Arbor, MI, USA, 1975.
46. Gen, M.; Cheng, R. *Genetic Algorithms and Engineering Optimization*; John Wiley & Sons: Hoboken, NJ, USA, 1999; Volume 7.
47. Deb, K. *Multi-Objective Optimization Using Evolutionary Algorithms*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2001.
48. Pinciroli, C.; Trianni, V.; O'Grady, R.; Pini, G.; Brutschy, A.; Brambilla, M.; Mathews, N.; Ferrante, E.; Di Caro, G.; Ducatelle, F.; et al. ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intell.* **2012**, *6*, 271–295. [CrossRef]
49. Dorigo, M.; Floreano, D.; Gambardella, L.M.; Mondada, F.; Nolfi, S.; Baaboura, T.; Birattari, M.; Bonani, M.; Brambilla, M.; Brutschy, A.; et al. Swarmanoid: A Novel Concept for the Study of Heterogeneous Robotic Swarms. *IEEE Robot. Autom. Mag.* **2013**, *20*, 60–71. [CrossRef]

50. Varrette, S.; Bouvry, P.; Cartiaux, H.; Georgatos, F. Management of an academic HPC cluster: The UL experience. In Proceedings of the 2014 International Conference on High Performance Computing & Simulation (HPCS), Bologna, Italy, 21–25 July 2014; IEEE: Bologna, Italy, 2014; pp. 959–967. [CrossRef]
51. Sheskin, D.J. *Handbook of Parametric and Nonparametric Statistical Procedures*; CRC Press: New York, NY, USA, 2003.
52. López-Ibáñez, M.; Paquete, L.; Stützle, T. Exploratory analysis of stochastic local search algorithms in biobjective optimization. In *Experimental Methods for the Analysis of Optimization Algorithms*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 209–222. [CrossRef]