Optimizing the Performance of an Unpredictable UAV Swarm for Intruder Detection

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

[1] Interdisciplinary Centre for Security, Reliability and Trust (SnT),
University of Luxembourg, Esch-sur-Alzette, Luxembourg
{daniel.stolfi,matthias.brust,gregoire.danoy,pascal.bouvry}@uni.lu
[2] FSTM/DCS, University of Luxembourg, Esch-sur-Alzette, Luxembourg

Abstract. In this paper we present the parameterisation and optimisation of the CACOC (Chaotic Ant Colony Optimisation for Coverage) mobility model applied to Unmanned Aerial Vehicles (UAV) in order to perform surveillance tasks. The use of unpredictable routes based on the chaotic solutions of a dynamic system as well as pheromone trails improves the area coverage performed by a swarm of UAVs. We propose this new application of CACOC to detect intruders entering an area under surveillance. Having identified several parameters to be optimised with the aim of increasing intruder detection rate, we address the optimisation of this model using a Cooperative Coevolutionary Genetic Algorithm (CCGA). Twelve case studies (120 scenarios in total) have been optimised by performing 30 independent runs (360 in total) of our algorithm. Finally, we tested our proposal in 100 unseen scenarios of each case study (1200 in total) to find out how robust is our proposal against unexpected intruders.

Keywords: Swarm robotics · Mobility model · Unmanned Aerial Vehicle · Evolutionary Algorithm · Surveillance

1 Introduction

Nowadays, one of the most common scenarios for Unmanned Aerial Vehicles (UAV) is the surveillance of exclusion areas such as army bases, private facilities or around prisons. In this scenario, UAVs equipped with cameras are used to explore a specific area in order to keep out unwelcome visitors [9]. Having this in mind, there is a need for intelligent surveillance trajectories [4] to prevent intruders from predicting the routes of UAVs and easily avoiding them as they move through the exclusion zone.

In [13] a mobility model for generating unpredictable trajectories for UAV swarms is proposed. This model, called Chaotic Ant Colony Optimisation for Coverage (CACOC), uses chaotic solutions of a dynamical system and

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 37–48, 2020.
https://doi.org/10.1007/978-3-030-41913-4_4

pheromones for guiding UAVs as well as improving the coverage of a given area using a multi-level swarm of collaborating UAVs.

CACOC relies on a set of parameters that can influence the vehicle's behavior and, consequently, the coverage performance. We propose in this paper an improvement of CACOC by calculating an optimised parameter set for this mobility model using an evolutionary bioinspired approach, with the aim of increasing the chances of spotting intruders in the area under surveillance, using a reduced number of UAVs and unpredictable trajectories. This is a novel use of CACOC since it has never been used for target detection.

The remainder of this paper is organised as follows. After reviewing the literature in the next section, the CACOC mobility model is explained and its parameters are discussed in Sect. 3. Section 4 focuses on the proposed optimisation algorithm. In Sect. 5 we present the characteristics of our case studies and the simulation environment. Section 6 focuses on our experimental results. And finally, in Sect. 7, conclusions and future work are given.

2 Literature Review

There are some research works which address route optimisation and surveillance using UAVs. In [12] a cooperative algorithm for optimizing UAVs routes is proposed where agents share information for the benefit of the team while searching for a target in minimum time. The authors performed simulations to test their proposal achieving improvements over traditional implementations. In [7] a Genetic Algorithm (GA) is proposed to optimise the parameters of a swarm of robots, with the objective of improving the mapping of the environment. By changing those parameters, the authors modify how agent-modeled ants travel from nest and use pheromone communication to improve foraging success.

In [17] a swarm of UAVs is optimised to improve target detection and tracking, map coverage, and network connectivity. They compare their proposed model, called Dual-Pheromone Clustering Hybrid Approach (DPCHA) with other approaches to obtain around 50% improvement in map coverage. In [2] a decentralised mobility model for UAV fleets based on Ant Colony Optimisation (ACO) is presented. It relies on attractive and repulsive pheromones to detect and track a maximum number of targets to perform surveillance and tracking missions. Attractive pheromones are used to follow and track discovered targets, while repulsive pheromones are used to survey the area by repelling UAVs to less scanned cells.

In [16] the authors present a chaotic predator-prey biogeography-based optimisation (CPPBBO) method, integrating the chaos theory and the concept of predator-prey into the classical Biogeography-Based Optimisation (BBO) algorithm. They use it to solve the Uninhabited Combat Air Vehicle (UCAV) path planning problem, with the aim of ensuring the maximum safety of the calculated path with the minimum fuel cost. In [1] a surveillance system composed of a team of UAVs is proposed. This is an efficient distributed solution for area surveillance which uses long endurance missions and limited communication range.

All these proposals use mobility models different from CACOC and do not provide unpredictable routes to improve area surveillance and intruder detection as we propose in our study, where we combine a cooperative bio-inspired approach with chaotic trajectories.

3 CACOC Mobility

CACOC (Chaotic Ant Colony Optimisation for Coverage) [13] is a mobility model based on chaotic dynamics and pheromone trails for improving area coverage using unpredictable trajectories. In spite of being unpredictable, CACOC's trajectories are also deterministic, what is extremely valuable if there are communication issues, allowing the Ground Control Station (GCS) to know where the vehicles are at any time. Algorithm 1 shows the pseudocode of CACOC.

Algorithm 1. Chaotic Ant Colony Optimisation for Coverage (CACOC).

```
procedure CACOC
    current state ← "ahead"
    loop
        $\rho \leftarrow$ next value in first return map
        if no pheromone sensed in the neighbourhood then
            if $\rho < \frac{1}{3}$ then                         ▷ CROMM
                current state ← "right"
            else if $\rho < \frac{2}{3}$ then
                current state ← "left"
            else
                current state ← "ahead"
            end if
        else
            if $\rho < P_R$ then                          ▷ Pheromones
                current state ← "right"
            else if $\rho < P_R + P_L$ then
                current state ← "left"
            else
                current state ← "ahead"
            end if
        end if
        move according to the current state
    end loop
end procedure
```

First, the next value in the first return map ρ describing a chaotic system (chaotic attractor obtained by solving an ordinary differential equations system, see [14]) is used to replace the random part of the mobility model. If there is no pheromone in the UAV's neighborhood, the next movement direction is given by CROMM (Chaotic Rössler Mobility Model) [13]. CROMM is an asymmetric mobility model which uses the first return map to calculate the UAV's next movement direction. It is a purely chaotic mobility model which does not use pheromones and the *current state* for each UAV is obtained according to an equally split partition as explained in [13]. Continuing with Algorithm 1, if pheromone trails are detected, they are used as repellers and the next action is calculated according to the probabilities shown in Table 1 [13].

Table 1. Pheromone action table.

Probability of action:	Left	Ahead	Right
	$P_L = \frac{total-left}{2\times total}$	$P_A = \frac{total-ahead}{2\times total}$	$P_R = \frac{total-right}{2\times total}$

When using pheromone repellers, UAVs are better spread in the area avoiding visiting the same spots too frequently. As pheromone trails evaporate, a UAV will eventually visit again the same region of the map. This is an intended behaviour since these UAVs are not mapping the area but performing surveillance tasks.

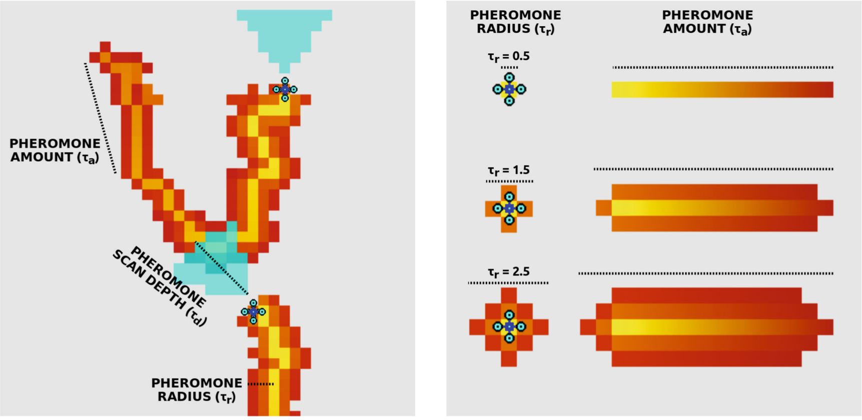
PHEROMONE AMOUNT (τ_a)
PHEROMONE SCAN DEPTH (τ_d)
PHEROMONE RADIUS (τ_r)
PHEROMONE RADIUS (τ_r)
PHEROMONE AMOUNT (τ_a)
$\tau_r = 0.5$
$\tau_r = 1.5$
$\tau_r = 2.5$

Fig. 1. Three pheromone parameters proposed for CACOC.

We propose three parameters in CACOC which are used for adapting this model to different scenarios, number of vehicles, etc., with the aim of increasing the probability of detecting intruders. We have parameterised the amount of pheromones left by each vehicle (τ_a), the pheromone radius (τ_r) and maximum detection distance (τ_d) as shown in Fig. 1. The higher τ_a, the longer the pheromones remains in the map as they are subject to a decay rate which is fixed to one unit per simulation step (tick).

Table 2 shows the parameters defined for CACOC to be optimised by the proposed algorithm in order to detect the maximum number of intruders.

Table 2. Parameters proposed for CACOC.

Parameter	Symbol	Units	Range
Pheromone amount	τ_a	%	[1–100]
Pheromone radius	τ_r	Cells	[0.5–2.5]
Pheromone scan depth	τ_d	Cells	[1–10]

4 Cooperative Coevolutionary Genetic Algorithm

We propose a Cooperative Coevolutionary Genetic Algorithm (CCGA) to maximise the probability of detecting and intruder fostering the collaboration between the members of the swarm. Our approach is based on the CCGA-2 [11] proposed as an extension of the traditional GA with the aim of representing and solving more complex problems by explicitly modeling the coevolution of cooperating species (Fig. 2).

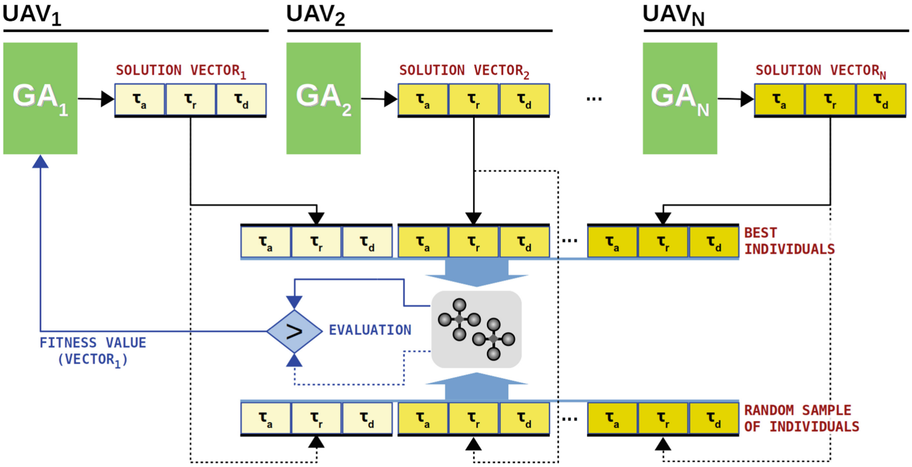

Fig. 2. Cooperative Coevolutionary Genetic Algorithm (CCGA). In this example, the SOLUTION VECTOR$_1$ of GA$_1$ is evaluated by completing the full configuration vector using the best individuals from the other GAs and a random sample of individuals from the other GAs, as well. The same process is followed to evaluate the rest of individuals in all the GAs' populations.

Each UAV has been assigned to a Genetic Algorithm (GA) to optimise its own set of parameters, i.e. τ_a, τ_r, and τ_d, as it is coded in each respective solution vectors using real numbers. Those GAs are identical and execute their own main loop until the evaluation stage where the full configuration vector is built using the best solution from the other GA's. Additionally, a second evaluation is performed using a random sample of individuals from the other GA's populations [11]. This technique reduces the convergence speed, fostering the populations' diversity.

Each GA is based on an Evolutionary Algorithm (EA) [5,8] which is an efficient method for solving combinatorial optimisation problems. EAs 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. This is a generational GA where an offspring of λ individuals is obtained from the population μ, so that the auxiliary population Q contains the same number of individuals (20 in our implementation) as the population P. The pseudocode of a GA is presented in Algorithm 2.

Algorithm 2. Pseudocode of the Genetic Algorithm (GA).

```
 1: procedure GA($N_i, P_c, P_m$)
 2:     $t \leftarrow \emptyset$
 3:     $Q(0) \leftarrow \emptyset$                                  ▷ Q=auxiliary population
 4:     $P(0) \leftarrow Initialisation(N_i)$                        ▷ P=population
 5:     while not $TerminationCondition()$ do
 6:         $Q(t) \leftarrow Selection(P(t))$
 7:         $Q(t) \leftarrow Crossover(Q(t), P_c)$
 8:         $Q(t) \leftarrow Mutation(Q(t), P_m)$
 9:         $Evaluation(Q(t))$
10:         $P(t+1) \leftarrow Replacement(Q(t), P(t))$
11:         $t \leftarrow t + 1$
12:     end while
13: end procedure
```

After initializing t and $Q(0)$ (lines 2 and 3), the GA generates $P(0)$ by using the *Initialisation* function (line 4). Then, the main loop is executed while the *TerminationCondition* is not fulfilled (in our case we stop after 30 generations). Following the main loop, the *Selection* operator is applied to populate $Q(t)$ using Binary Tournament [6] (line 6). After that, the *Crossover* operator is applied (line 7) and then, the *Mutation* operator slightly modifies the new offspring (line 8). Finally, after the *Evaluation* of $Q(t)$ (line 9), the new population $P(t+1)$ is obtained by applying the *Replacement* operator (line 10). In order to avoid population stagnation and preserve its diversity (and entropy), we have selected the best individual in $Q(t)$ to replace the worst one in $P(t)$ [3] if it detects more intruders (it has a better fitness value).

4.1 Operators

We have based our crossover operator and mutation operator on the ones proposed in [3] for solving continuous optimizing problems. The crossover operator is applied to each two individuals in the population (X and Y) with a crossover probability $P_c = 0.9$ calculated in our previous tests. First, a random integer M

between 1 and the length of the solution vector (L) is obtained as shown in Eq. 1 and used to calculate the value of both Δ_x and Δ_y, see Eq. 2. After that, all the configuration values in the solution vector beyond a randomly selected crossing point are changed according to Eq. 3.

$$M = randInt(1, L) \tag{1}$$

$$\Delta_x = \frac{X_i}{M}, \quad \Delta_y = \frac{Y_i}{M} \tag{2}$$

$$X\prime_i = X_i + \Delta_y - \Delta_x, \quad Y\prime_i = Y_i - \Delta_y + \Delta_x \tag{3}$$

The mutation operator [3] has also been adapted to our problem characteristics. In this case a new value M is calculated as in Eq. 1 and used to get the value of Δ (Eq. 4) taking into account the upper and lower bound of each configuration variable (Table 2). Then, with a mutation probability $P_m = \frac{1}{L}$ the value of Δ will we either subtracted from or added to the variable X_i according to a probability $P_d = \frac{1}{2}$ (equiprobable).

$$\Delta = \frac{UpBd(X_i) - LowBd(X_i)}{M} \tag{4}$$

$$X\prime_i = \begin{cases} X_i - \Delta & \text{if } P_d < 0.5 \\ X_i + \Delta & otherwise \end{cases} \tag{5}$$

4.2 Fitness Function

Our objective is maximizing the efficiency of the surveillance system, i.e. maximise the number of intruders detected. Therefore, the evaluation consists of obtaining the percentage of intruders successfully detected by the UAV swarm when its members are configured by the parameters in $\boldsymbol{x}$, during the analysis time (600 s). We also consider as a successful situation when the analysis time ends and an intruder has not reached its destination despite not having been detected by any UAV, see Eq. 6. This could happen since intruders are able to evade UAVs but they deviate from their original trajectory, run out of time, and never reach their destination. In order to increase the robustness of our proposal, we evaluate ten different scenarios ($\gamma = 10$) and obtain the fitness value using the Monte Carlo method [10]. As we are maximizing the average number of detections, the higher the value of $F(\boldsymbol{x})$, the better.

$$F(\boldsymbol{x}) = \frac{1}{\gamma} \sum_i \frac{\#\ of\ intruders_i - \#\ of\ intruders\ at\ destination_i}{\#\ of\ intruders_i} \tag{6}$$

5 Simulation Environment

We use a simulation environment in order to test our proposal and optimise the UAV's parameters. Each scenario is represented as a lattice of 100 by 100 cells (Fig. 3) where the UAV's can move following the mobility model (CACOC in this study) leaving pheromones behind while scanning the area under their detection area (calculated according to real camera specifications). The centre of the map contains the region to be protected where intruders wish to arrive. A swarm of UAVs will try to prevent that by using unpredictable trajectories. If an intruder is detected, it is removed from the scenario and counted as a success. At the end of a simulation, all the intruders which have not reached destination are also considered as a success despite the fact they have not been detected yet.

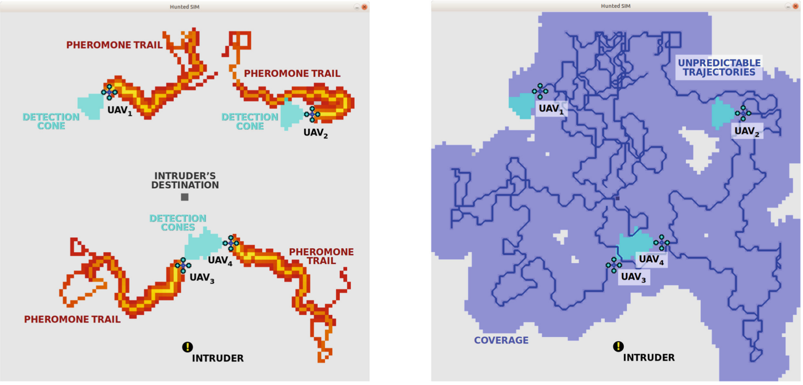

Fig. 3. Snapshots of our simulation environment: HUNTED SIM.

For this study the intruders' behavior has been modeled using a repelling force which makes each intruder try to avoid UAV's (Eq. 7) which competes with an attracting force towards destination (Eq. 9). The intruder's next movement (Eq. 10) will depend on the relative position of the destination given by $\boldsymbol{x}_a$ and the surrounding UAVs (if any) which are closer than the maximum distance D_{max}. These UAVs contribute to the repelling force $\boldsymbol{x}_r$ proportionally as given by δ_i (Eq. 8). Finally, the resulting moving direction is normalised to be scaled according to the movement speed of each intruder.

$$\boldsymbol{x}_{r(t+1)} = \sum_i [(\boldsymbol{uav}_i - \boldsymbol{x}_{(t)}) * \delta_i] \tag{7}$$

$$\delta_i = \begin{cases} D_{max} \times \|\boldsymbol{uav}_i - \boldsymbol{x}_{(t)}\|^{-1} & \text{if } \|\boldsymbol{uav}_i - \boldsymbol{x}_{(t)}\| < D_{max} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$\boldsymbol{x}_{a(t+1)} = \frac{\boldsymbol{dest} - \boldsymbol{x}_{(t)}}{\|\boldsymbol{dest} - \boldsymbol{x}_{(t)}\|} \tag{9}$$

$$\boldsymbol{x}_{(t+1)} = \frac{\boldsymbol{x}_{r(t+1)} + \boldsymbol{x}_{a(t+1)}}{\|\boldsymbol{x}_{r(t+1)} + \boldsymbol{x}_{a(t+1)}\|} \tag{10}$$

In this study the intruders move at the same speed as UAVs, have a detection (sight) angle of 180°, and are able to see UAVs up to 10 m away. We have set up 12 surveillance case studies in which there are 2, 4, 6, and 8 intruders being chased by 4, 8, 12, and 16 UAVs providing there are more UAVs than intruders. Following the pattern `UAVS.INTRUDERS` we have named each case study as 4.2, 8.2, 8.4, 8.6, 12.2, 12.4, 12.6, 12.8, 16.2, 16.4, 16.6, and 16.8. We have also defined ten scenarios for each case study to improve the robustness of the system. Scenarios differ in the arrival time of intruders and the position by which they enter the area (always by one border of the map), both of which have been chosen randomly. Since we set up an analysis time of 600 s the maximum arrival time is under 400 s.

6 Experimental Results

The experiments were conducted in two stages. First, we addressed the optimisation of each case study by performing 30 independent runs of CCGA-2 including ten scenarios each. The whole optimisation process needed 360 runs in total.

Table 3 shows the fitness value obtained for each case study and its optimization time. Since we have performed 30 independent runs because CCGA is non deterministic, we report the average, standard deviation, minimum, and maximum (best) fitness values achieved. It can be seen that the more UAVs in the map, the better, as expected.

Moreover, fitness values (success rate) decrease when there are more intruders trying to reach their destination, although a higher number of UAVs mitigates in part this matter, e.g. 12 UAVs are more successful in catching six intruders than 8 UAVs. The aforementioned tendencies can be also observed in Fig. 4.

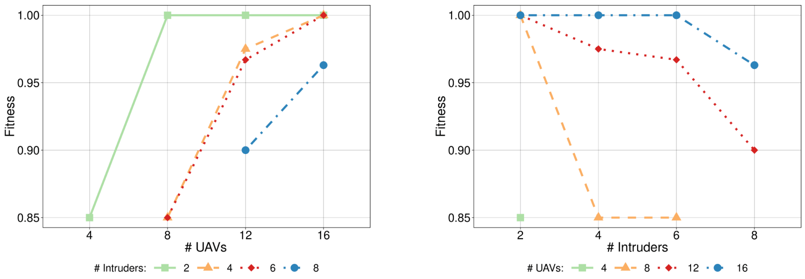

Fig. 4. Fitness value increases with the number of UAVs and it decreases when there are more intruders in the scenario.

The second stage consisted in testing the best configuration for each set of UAVs (4, 8, 12, and 16) on 100 unseen scenarios of each case study. We report in Table 4 the average success rate obtained. It can be seen that again the more UAVs, the better, so that 8 UAVs are in the 57%–60% success range, 12 UAVs are in 74%–80%, and 16 UAVs are around 84%–87% on average.

Table 3. Results of the optimisation process performed by CCGA-2 (30 runs).

Case study	Fitness				Time (Hours)
	Avg.	StdDev.	Min.	Max.	
4.2	0.710	0.064	0.550	0.850	1.0
8.2	0.905	0.038	0.850	1.000	2.9
8.4	0.808	0.031	0.750	0.850	3.1
8.6	0.775	0.035	0.717	0.850	3.0
12.2	0.982	0.025	0.950	1.000	6.1
12.4	0.926	0.021	0.900	0.975	6.1
12.6	0.888	0.023	0.850	0.967	6.0
12.8	0.863	0.018	0.825	0.900	6.1
16.2	1.000	0.000	1.000	1.000	10.2
16.4	0.983	0.012	0.975	1.000	10.4
16.6	0.960	0.016	0.933	1.000	10.5
16.8	0.945	0.012	0.925	0.963	10.4

Table 4. Average detection percentage after testing of the best configurations obtained in 100 scenarios of each case study.

UAVs	Intruders				Total
	2	4	6	8	
4	16.0%	—	—	—	16.0%
8	57.5%	57.3%	59.0%	—	58.2%
12	79.5%	75.8%	74.3%	76.0%	75.8%
16	86.0%	87.0%	84.0%	85.1%	85.3%

Figure 5 shows the distribution of these values for each case study, i.e. 1200 different scenarios. It can be seen that there are scenarios in which all the intruders were spotted (100% success) while in other, no one was detected (0% success). We went deeper in our analysis focused on the 0% success cases to discover that intruders managed to dodge the UAVs by going backwards and trying to move forward again avoiding the UAVs in the neighbourhood, until they arrived to destination. All in all, no intruders were detected in 72% of scenarios in 4.2, 16% in 8.2, 4% in 8.4, 4% in 12.2 and 2% in 16.2.

We have conducted our experimentation using computing nodes equipped with Xeon Gold 6132@2.6 GHz and 128 GB of RAM. It took about 80 h of parallel runs (90 equivalent days).

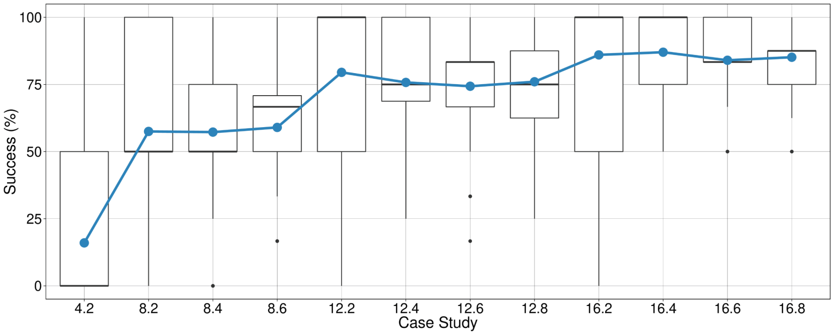

Fig. 5. Average success rate for each case study (100 unseen scenarios each).

7 Conclusion

In this paper we have proposed new features for the CACOC (Chaotic Ant Colony Optimisation for Coverage) mobility model to be used as part of an intruder detection system for the first time. We have optimised the newly proposed parameters using an Cooperative Coevolutionary Genetic Algorithm (CCGA) to maximise the intruder detection rate when the swarm of UAVs is performing surveillance tasks.

We have detected up to 100% of intruders during the optimisation stage and after testing the best configurations achieved in 1200 scenarios we have observed detection rates up to 87%. These results show that the parameters selected can be optimised to modify the swarm behaviour in order to improve the detection of intruders. Moreover, the coevolutionary strategy allowed the individual configuration of each UAV in the swarm, which could be observed in the robustness of the system when tested against new unseen scenarios.

As future work we want to analyse the predators' behaviour in order to improve further our proposal. We plan to address a coevolutionary approach optimising also the intruders, implementing a competitive evolution of both species following game theory's rules. Also, we wish to evolve full configuration vectors using a GA and compare its results with the CCGA ones.

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 [15] – see https://hpc.uni.lu.

References

1. Acevedo, J.J., Arrue, B.C., Maza, I., Ollero, A.: Cooperative large area surveillance with a team of aerial mobile robots for long endurance missions. J. Intell. Robot. Syst. **70**(1–4), 329–345 (2013)
2. Atten, C., Channouf, L., Danoy, G., Bouvry, P.: UAV fleet mobility model with multiple pheromones for tracking moving observation targets. In: Squillero, G., Burelli, P. (eds.) EvoApplications 2016. LNCS, vol. 9597, pp. 332–347. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31204-0_22
3. Chelouah, R., Siarry, P.: Continuous genetic algorithm designed for the global optimization of multimodal functions. J. Heuristics **6**(2), 191–213 (2000)
4. Galceran, E., Carreras, M.: A survey on coverage path planning for robotics. Robot. Auton. Syst. **61**(12), 1258–1276 (2013)
5. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning, 1st edn. Addison-Wesley Longman Publishing Co., Inc., Boston (1989)
6. Goldberg, D.E., Deb, K.: A comparative analysis of selection schemes used in genetic algorithms. Found. Genet. Algorithms **1**, 69–93 (1991)
7. Hecker, J.P., Letendre, K., Stolleis, K., Washington, D., Moses, M.E.: *Formica ex Machina*: ant swarm foraging from physical to virtual and back again. In: Dorigo, M., et al. (eds.) ANTS 2012. LNCS, vol. 7461, pp. 252–259. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32650-9_25
8. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT press, Cambridge (1992)
9. McNeal, G.S.: Drones and the future of aerial surveillance. George Wash. Law Rev. Arguendo **84**(2), 354 (2016)
10. Metropolis, N., Ulam, S.: The Monte Carlo method. J. Am. Stat. Assoc. **44**(247), 335–341 (1949)
11. Potter, M.A., De Jong, K.A.: A cooperative coevolutionary approach to function optimization. In: Davidor, Y., Schwefel, H.-P., Männer, R. (eds.) PPSN 1994. LNCS, vol. 866, pp. 249–257. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-58484-6_269
12. Riehl, J.R., Collins, G.E., Hespanha, J.P.: Cooperative search by UAV teams: a model predictive approach using dynamic graphs. IEEE Trans. Aerospace Electron. Syst. **47**(4), 2637–2656 (2011)
13. Rosalie, M., Danoy, G., Chaumette, S., Bouvry, P.: Chaos-enhanced mobility models for multilevel swarms of UAVs. Swarm Evol. Comput. **41**(November 2017), 36–48 (2018)
14. Rosalie, M., Letellier, C.: Systematic template extraction from chaotic attractors: II. genus-one attractors with multiple unimodal folding mechanisms. J. Phys. A: Math. Theor. **48**(23), 235101 (2015)
15. 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 2014), pp. 959–967. IEEE, Bologna, July 2014
16. Zhu, W., Duan, H.: Chaotic predator-prey biogeography-based optimization approach for UCAV path planning. Aerospace Sci. Technol. **32**(1), 153–161 (2014)
17. Zurad, M., et al.: Target tracking optimization of UAV swarms based on dual-pheromone clustering. In: 2017 3rd IEEE International Conference on Cybernetics (CYBCONF), pp. 1–8, no. October. IEEE, June 2017