



UAV-UGV-UMV Multi-Swarms for Cooperative Surveillance

Daniel H. Stolfi^{1*}, Matthias R. Brust¹, Grégoire Danoy^{1,2} and Pascal Bouvry^{1,2}

¹Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Esch-sur-Alzette, Luxembourg,

²FSTM/DCS, University of Luxembourg, Esch-sur-Alzette, Luxembourg

In this paper we present a surveillance system for early detection of escapers from a restricted area based on a new swarming mobility model called CROMM-MS (Chaotic Rössler Mobility Model for Multi-Swarms). CROMM-MS is designed for controlling the trajectories of heterogeneous multi-swarms of aerial, ground and marine unmanned vehicles with important features such as prioritising early detections and success rate. A new Competitive Coevolutionary Genetic Algorithm (CompCGA) is proposed to optimise the vehicles' parameters and escapers' evasion ability using a predator-prey approach. Our results show that CROMM-MS is not only viable for surveillance tasks but also that its results are competitive in regard to the state-of-the-art approaches.

Keywords: swarm robotics, mobility model, UAV, UGV, UMV, competitive coevolutionary genetic algorithm, surveillance system, parameter optimisation

OPEN ACCESS

Edited by:

Savvas Loizou,
Cyprus University of Technology,
Cyprus

Reviewed by:

Hassène Grilli,
University of Tunis, Tunisia
Philip M Dames,
Temple University, United States

*Correspondence:

Daniel H. Stolfi
daniel.stolfi@uni.lu

Specialty section:

This article was submitted to
Multi-Robot Systems,
a section of the journal
Frontiers in Robotics and AI

Received: 13 October 2020

Accepted: 08 January 2021

Published: 19 February 2021

Citation:

Stolfi DH, Brust MR, Danoy G and
Bouvry P (2021) UAV-UGV-UMV Multi-
Swarms for Cooperative Surveillance.
Front. Robot. AI 8:616950.
doi: 10.3389/frobt.2021.616950

1 INTRODUCTION

The use of different types of vehicles arranged in swarms is a promising strategy (Girma et al., 2020) which allows to take advantage of the capabilities of each member while minimising their limitations. These heterogeneous swarms are based on collective behaviors of social insects where collaborations emerge to achieve a common goal (Jeong et al., 2019) while avoiding no-fly zones, extending battery life, or braving inclement weather.

Unmanned Aerial Vehicles (UAVs), also known as drones, are widely used in many applications nowadays (McNeal, 2016). In surveillance scenarios (Brust et al., 2017), they can explore different areas at a relatively high speed using their cameras, featuring excellent communication capabilities (Batista da Silva et al., 2017). On the other hand, UAVs have a reduced flight time, cannot carry larger payloads, and have big troubles to stand in a static position when there are strong winds.

Unmanned Ground Vehicles (UGVs) are able to work over a wide variety of terrains, sharing space with humans, searching for targets while avoiding obstacles (Brust and Strimbu, 2015). UGVs are good candidates for a joint usage with UAVs, complementing each other to act in different terrain characteristics (Stolfi et al., 2020c). Although UGVs are slower than UAVs and have a limited communication range (line of sight, obstacles, etc.), they have a higher autonomy and are frequently used to support UAVs providing a moving recharging station and transportation between mission objectives (Waslander, 2013).

Unmanned Surface Vehicles (USVs), which include Unmanned Marine Vehicles (UMVs), have been proposed as critical components of the future naval forces (Yan et al., 2010) to perform missions such as mine countermeasures, maritime security, and maritime interdiction operations support, among others. They are thought to improve existing naval capabilities offering a reduction in both operational time and cost (Costanzi et al., 2020).

Unpredictability is an essential feature when calculating trajectories for robots which have to perform risky missions such as surveillance, patrolling, or searching for mines. Mobility models

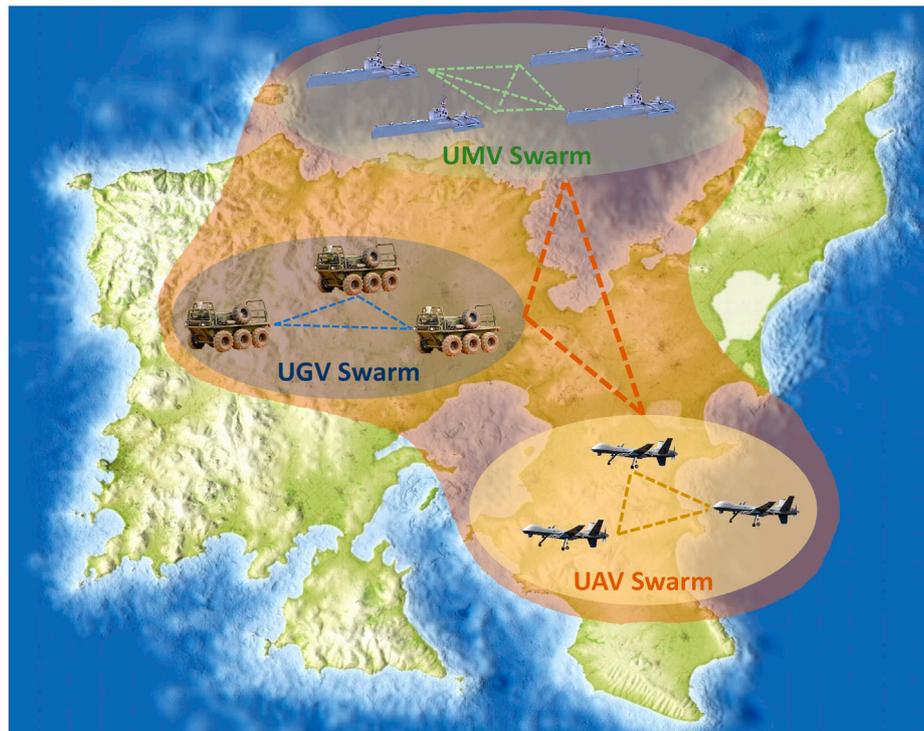


FIGURE 1 | Inter-swarm collaborations as proposed in the HUNTED project – see <https://hunted.gforge.uni.lu>.

based on chaotic systems present such desired unpredictability (Iba and Shimonishi, 2011), only being sensitive to the initial conditions and the implementation which can be made via software to be used in a simulator (Rosalie et al., 2018) or implemented as part of the robot electronics, featuring a true random bit generator obtained from a multi-scroll attractor, to provide waypoints for UAVs (Volos et al., 2012).

Our proposal consists of a novel multi-swarm surveillance system (Figure 1) where UAVs, UGVs, and UMVs collaborate to achieve early detection of escapers from a restricted area. To this end, we introduce an extension of CROMM (Chaotic Rössler Mobility Model), initially designed for homogeneous vehicles and area coverage, for now addressing heterogeneous multi-swarms and spotting escaper in surveillance scenarios. We aim to exploit the described best features of each vehicle class to achieve better results than homogeneous solutions. The main contributions of this paper are:

1. A new mobility model called CROMM-MS (Chaotic Rössler Mobility Model for Multi-Swarms) to control UAVs trajectories with the aim of maximising early escaper detection.
2. A predator-prey approach to train and improve this surveillance system.
3. A Competitive Coevolutionary Genetic Algorithm (CompGA) specially designed for optimising the vehicles parameters and improving escapers to be valid evaluators.

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 our approach is presented. The experimental results are in Section 4. And finally, Section 5 brings discussion and future work.

2 RELATED WORK

Multi-pursuer and multi-evader games have received great attention in the literature. In (Makkapati and Tsiotras, 2019, 2020) two single navigation laws (constant bearing and pure pursuit) are proposed using Apollonius circles and curves. Additionally, a task allocation algorithm is proposed for the pursuers to solve the problem in finite time for any number of pursuers and evaders. A Graph Neural Network trained with Reinforcement Learning was used in (Deka and Sycara, 2020) to obtain complex strategies for two teams of agents. The authors also proposed a mixed cooperative-competitive multi agent environment called FortAttack to test their proposal. After the training process, they achieved highly competitive, emergent heterogeneous behavior between the homogeneous agents. In (Pierson et al., 2017) an algorithm for cooperative pursuit of multiple evaders using multiple pursuers is proposed. It uses a global area minimisation strategy based on the Voronoi tessellation of the environment to capture all the evaders in finite time. Two successful experiments were conducted in a 4x3-metre real environment using four pursuers against four autonomous evaders and also against one human-controlled evader and three autonomous evaders.

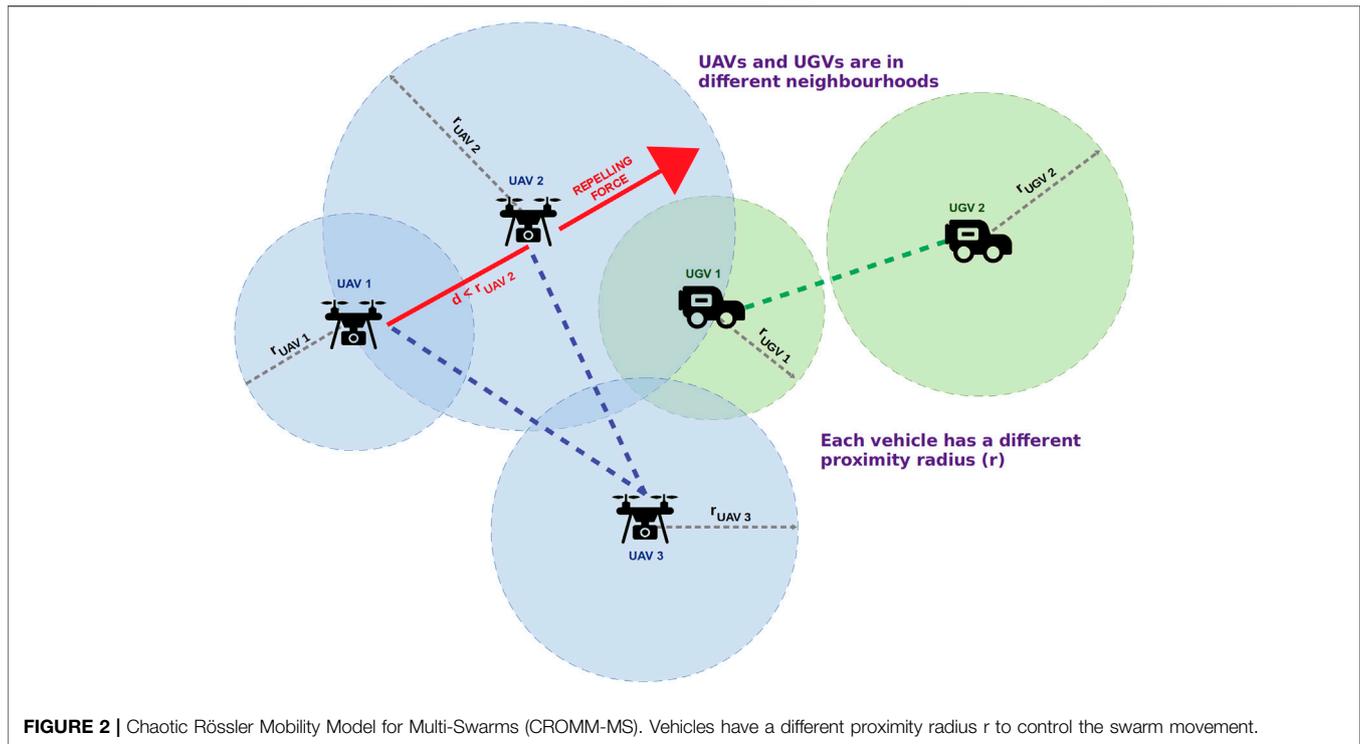


FIGURE 2 | Chaotic Rössler Mobility Model for Multi-Swarms (CROMM-MS). Vehicles have a different proximity radius r to control the swarm movement.

Path planning algorithms using chaotic dynamics have been used to get unpredictable trajectories. In (Petavratzis et al., 2019) a modification of the well-known logistic map is used to generate a chaotic pseudorandom bit generator (CPRBG) to produce a bit sequence. This sequence is then used to control a robot which moves by a grid in four or eight different directions. A simple, memory efficient, pheromone model is also proposed to improve the achieved coverage level. In (Moysis et al., 2020) an extension of the previous paper is discussed to avoid the use of a CPRBG taking into account the limited onboard memory and microcontrollers. This new approach is based on getting chaotic tactics by applying a logistic map and a modulo operator. In this article, pheromones are also proposed to increase coverage level and reduce the number of revisits in previous cells.

Coevolutionary algorithms are a good choice to solve problems involving cooperation or competition between different population of individuals. In (Tiguercha et al., 2014) a competitive coevolutionary algorithm is presented to model the interactions of several agents to find the optimal binding strategies in a deregulated electricity market. Each agent is modeled as an adaptive evolutionary agent that acts strategically in order to maximise their profits. A competitive coevolutionary search to the code-smells detection problem is proposed in Boussaa et al. (2013). The authors present two populations that compete one against the other. The first one generates a set of detection rules to maximise the coverage of code-smell examples, while the second population focuses on maximising the number of code smells that cannot be detected. In (Wiegand et al., 2001) an empirical study about cooperative coevolutionary algorithms is presented. Three methods for

assigning fitness values based on its collaborations are proposed as well as different collaboration mechanisms using collaborator pools of different sizes.

Several recent research works address cooperative heterogeneous multi-robot systems to perform a variety of tasks (Huang et al., 2019; Rizk et al., 2019). In (Vu et al., 2018) several UAV-UGV cooperation tasks for applications in the field of architecture are presented. Focusing on each vehicle characteristics, UGVs are thought to exploit their working autonomy and high level of interaction with the user while UAVs are more appropriate for communications and tracking support. A probabilistic and scalable new strategy to solve the multi-robot patrol problem is proposed in (Portugal and Rocha, 2016). It uses Bayesian decision-making combined with adaptive learning to achieve intelligent patrolling routes which are tested in a simulation environment as well as in the real world. In (Jayavelu et al., 2018) UGVs are proposed as mobile refuelling and maintenance stations for UAVs. The authors present a framework to calculate the optimal number of UGVs and their location according to the density and position of the moving UAVs. A three-layer surveillance system is proposed in Lee et al., (2019) where UGVs are used as ground-level proximity sensors, UAVs in the second layer use vision sensors, and in the upper layer aerostats (hot-air balloons) provide a broader view of the surveillance area. The authors use an extended DDDAMS (Dynamic-data-driven Adaptive Multi-scale Simulation) using a real-time detection and classification algorithm to predict the target's location based on a human behavior model. A cooperative exploration solution for search and rescue application in a damaged building is presented in (Hood et al., 2017). It consists of a UGV which navigates through the free indoor

TABLE 1 | Parameters of each escaper (prey).

Parameter	Symbol	Units	Values	Type
Border	ϵ_b	N,S,E,W	(0 – 3)	Integer
Coordinate	ϵ_c	Coordinate	(0 – 399)	Integer
Speed	ϵ_s	Simulation ticks	(2 – 6)	Integer
Time	ϵ_t	Simulation ticks	(300 – 1200)	Integer
Avoidance	ϵ_a	—	(0.00 – 2.00)	Real

space and a UAV providing enhanced situational awareness. In (Arbanas et al., 2018) a decentralised task planning and coordination framework is proposed. It includes a symbiotic aerial vehicle-ground vehicle robotic team where UAVs are used for aerial manipulation tasks, while UGVs assist them by providing safe landing areas and transport. Additionally, UAVs also helps UGVs to negotiate obstacles from their vantage point.

Our proposal does not involve two but three different types of vehicles arranged in nine swarms where each group supports its counterparts performing surveillance tasks. Predators do not have any information about preys and preys are aware of predators only when they are in their vision range. We use a competitive predator-prey approach and optimise not only the autonomous vehicles but also escapers from a restricted area in order to present smarter opponents to be spotted as soon as possible.

3 MATERIALS AND METHODS

We present in this paper an autonomous intelligent surveillance system for detection of escapers breaking out of a restricted area.

It is composed by swarms of unmanned autonomous vehicles of different types which patrol the surveillance area in order to spot individuals before they reach the map borders. We follow a predator-prey approach optimising all the entities involved in each escape scenario using an evolutionary bio-inspired technique as described in the following sections.

3.1 Swarms of Autonomous Vehicles (Predators)

Our surveillance system is composed of three types of autonomous vehicles each equipped with different detection capabilities:

- UGVs: medium speed, short detection range, 90-degree vision.
- UAVs: high speed, medium detection range, zenithal camera.
- UMVs: low speed, high detection range, 360-degree detection.

We propose CROMM-MS (Chaotic Rössler Mobility Model for Multi-Swarms) an extension of the CROMM (Rosalie et al., 2018) mobility model adapted to heterogeneous multi-swarms of unmanned vehicles moving throughout large scenarios. CROMM is a pure chaotic mobility model where the mobility decisions are taken according to the first return map of a Rössler system (Rössler, 1976). The first return map provides an unpredictable sequence of values $\rho \in [0 - 1]$, that are used to decide the next moving direction of a vehicle following a probability partition. Consequently, if $\rho < 1/3$, the vehicle turns

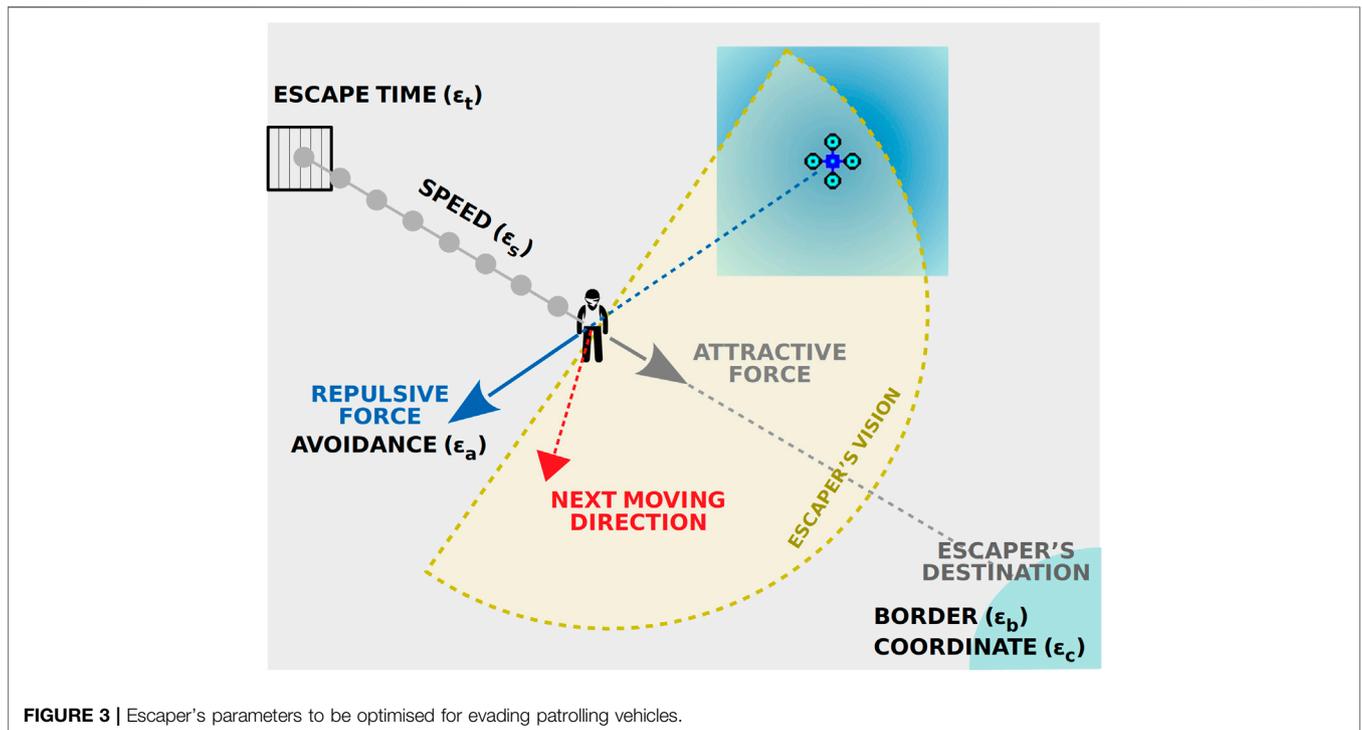
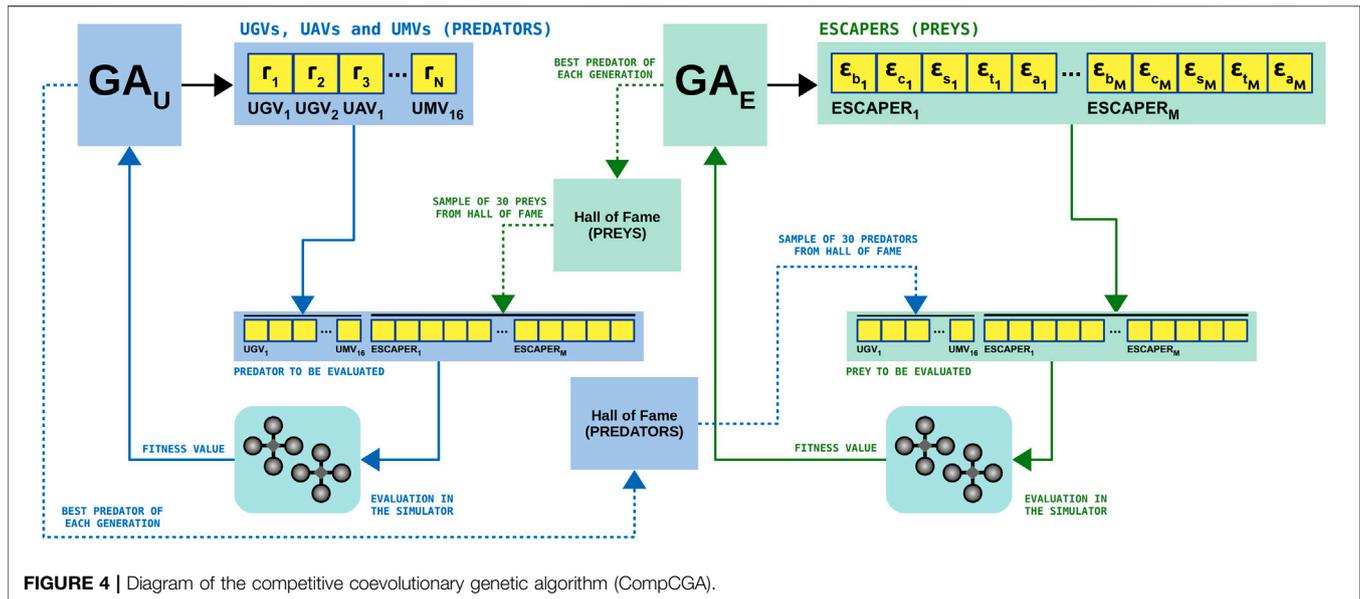


FIGURE 3 | Escaper's parameters to be optimised for evading patrolling vehicles.



right ($-\pi/4$); if $1/3 \leq \rho < 2/3$, the vehicle turns left ($\pi/4$); and if $\rho \geq 2/3$, it keeps moving ahead.

Unlike CROMM, when there are other vehicles in the neighborhood given by the proximity radius r , our approach uses a repelling vector calculated taking into account the other vehicles, to decide the next moving direction as shown in **Figure 2**. If there are no other vehicles in the neighborhood, CROMM-MS works as a pure chaotic mobility model (like CROMM) where the mobility decisions are taken according to the first return map ρ . CROMM-MS runs in each vehicle i using a different proximity radius r_i to be optimised taking into account each vehicle's characteristics, with the aim of better spreading the swarms' members throughout the surveillance scenario, improving coverage and detection rates, without creating big gaps that could be exploited by escapers.

Algorithm 1 shows the pseudocode of CROMM-MS. First, the current angle for each vehicle in the swarm respect to the others is obtained, and its proximity radius r is checked against the rest of vehicles of the same type. If some vehicles are closer than r , i.e. $V \neq \emptyset$, the new moving direction (angle) is calculated according to the respective repelling forces ($\vec{\Delta}$). Otherwise, pure chaotic is used, based on the value of ρ and the aforementioned probability partition.

Algorithm 1 Pseudocode of CROMM-MS.

```

procedure CROMM-MS
  for all  $u \in U$  do  $\triangleright$  For each vehicle in the swarm
    angle  $\leftarrow$  CurrentAngle( $u$ )
     $V \leftarrow v \in U : v \neq u \wedge \text{Distance}(v, u) < r$ 
    if  $V \neq \emptyset$  then  $\triangleright$  If there are vehicles closer than  $r$ 
       $\vec{\Delta} \leftarrow \sum (\vec{u} - \vec{v})$ 
      angle  $\xleftarrow{\epsilon} a \tan(\Delta_y/\Delta_x)$ 
    else  $\triangleright$  CROMM
       $\rho \leftarrow$  next value in first return map
      if  $\rho < 1/3$  then
        turn  $\leftarrow -\pi/4$   $\triangleright$  Turn right

```

```

else if  $\rho < 2/3$  then
  turn  $\leftarrow \pi/4$   $\triangleright$  Turn left
else
  turn  $\leftarrow 0$   $\triangleright$  Go straight on
end if
  angle  $\leftarrow$  angle + turn
end if
  Move( $u$ , angle)  $\triangleright$  The predator moves
end for
end procedure

```

3.2 Escapers (Preys)

We have also designed an escaper (prey) mobility model as a complementary component of our system. It consists of a series of parameters to be optimised simultaneously with the vehicles (predators). Each escaper has an *escape time* (ϵ_t) which defines when it breaks out from the facilities in the center of the map and tries to reach one of the *borders* (ϵ_b) at a predefined *coordinate* (ϵ_c). Finally, the running *speed* (ϵ_s) and the intensity of the *avoidance* (ϵ_a) manoeuvres (repulsive force) are also parameterised aiming to present difficult adversaries to our surveillance solution.

Table 1 shows the parameter list of each escaper and in **Figure 3** the forces involved in the escaper's mobility decision are illustrated. These degrees of freedom provide escapers the ability of deciding when attempting to run away, the less populated region of the map for that escape attempt, and how to react more efficiently to the predators' menace. All these parameters are optimised by our competitive algorithm as described in the following section.

3.3 Competitive Coevolutionary Genetic Algorithm (CompCGA)

We propose a Competitive Coevolutionary Genetic Algorithm (CompCGA) for optimising the proximity radius (r) of each

TABLE 2 | Parameters of the proposed CompCGA. L is the length of the configuration vector.

Parameter	Value
Population (μ)	20
Offspring (λ)	20
# Evaluations	20,000
Crossover probability (P_c)	1.0
Mutation probability (P_m)	1/L

autonomous vehicle and the intruders' parameters in a competitive way following a predator-prey approach. We have taken some initial steps developing our CompCGA in (Stolfi et al., 2020b) optimising a homogeneous swarm of UAVs and a number of intruders in a surveillance scenario. We have now adapted our algorithm and improved it, using two Hall of Fame tables to be sure that the best specimens are used to evaluate their counterparts as well as keeping a memory of the former good candidates. This new version of CompCGA needs more simulations to evaluate an individual but ensures that each fitness value represents how good is a given configuration at defeating several competitors, instead of just using the last best competitor from the other population.

We have used a Genetic Algorithm (GA) to address each parameter set individually as depicted in **Figure 4**. Genetic Algorithms (Goldberg, 1989; Holland, 1992) are efficient methods for solving combinatorial optimisation problems. They 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. In this work we consider generational GAs 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 case) as the population P . Two identical GAs (GAU for vehicles and GAE for escapers) are used in the CompCGA to perform the coevolution of their populations. The pseudocode of one of these GAs is presented in Algorithm 2.

Algorithm 2 Pseudocode of each Genetic Algorithm (GA).

```

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

After initialising t and $Q(0)$, the GA generates the initial population $P(0)$ by using the *Initialisation* function. The main loop is executed while the *TerminationCondition* is not fulfilled (we stop after 20,000 evaluations). Into the main loop, the *Selection* operator is applied to populate $Q(t)$ using Binary Tournament (Goldberg and Deb, 1991). In the following lines, our *Crossover* and *Mutation* operators (Stolfi et al., 2020a, Stolfi et al., 2020d) are applied. The former with the aim of widely exploring the search space and the latter to make small modifications to each individual (solution vector)

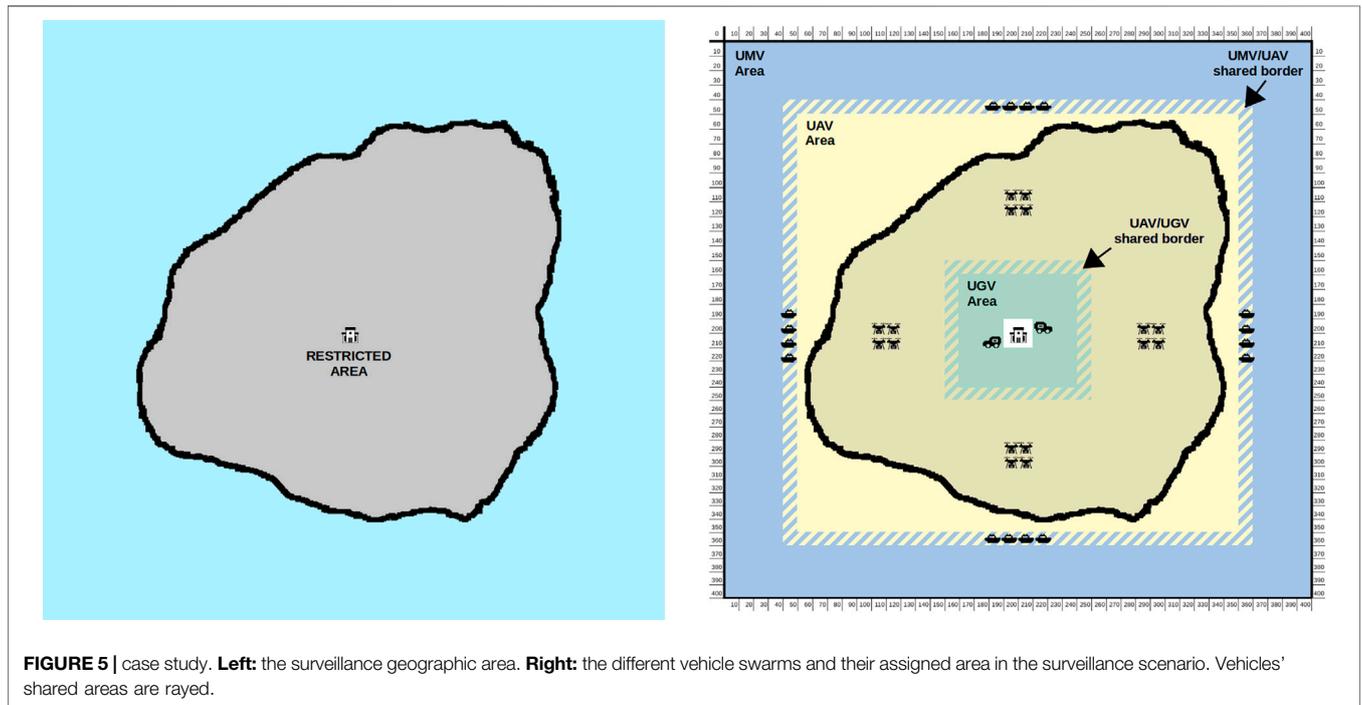


FIGURE 5 | case study. **Left:** the surveillance geographic area. **Right:** the different vehicle swarms and their assigned area in the surveillance scenario. Vehicles' shared areas are rayed.

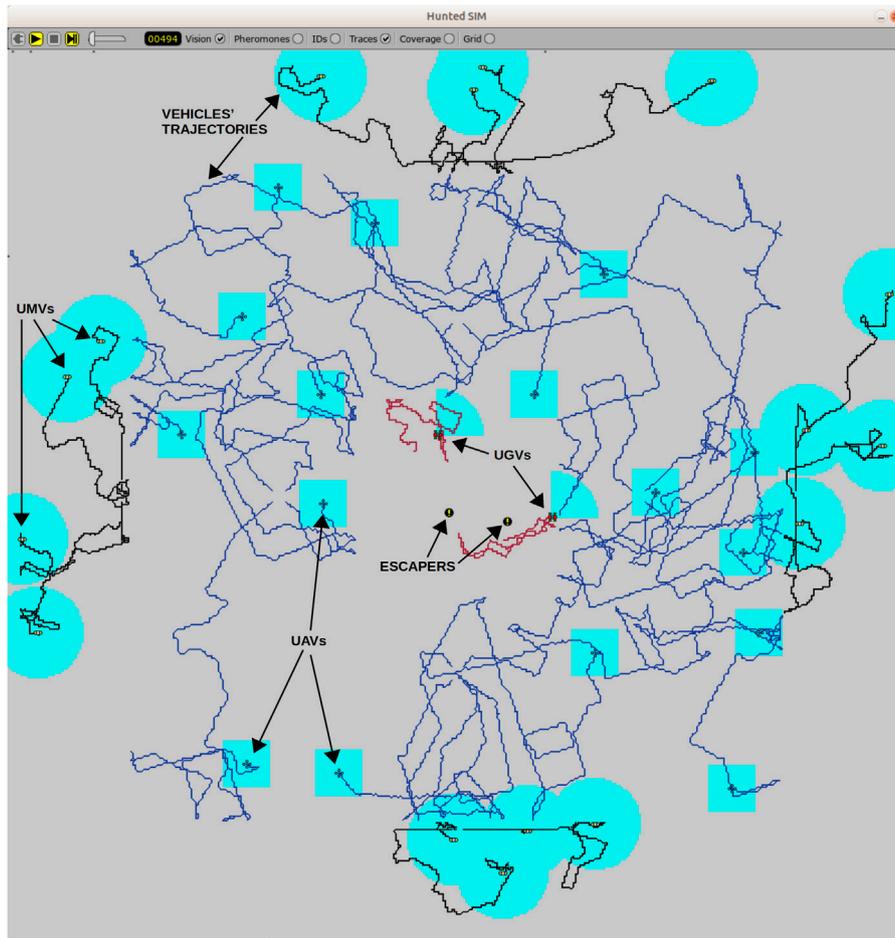


FIGURE 6 | Snapshot of the Hunted Sim where each swarm is spread by the surveillance map. Vehicles trajectories can be seen as well as their detection area. Near to the central area, two escapers are starting their run away.

of the offspring. Finally, after the *Evaluation* of $Q(t)$, the new population $P(t + 1)$ is obtained by applying the *Replacement* operator. 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 $P(t)$ (Stolfi et al., 2020a, Stolfi et al., 2020d) if it has a better fitness value.

Each individual is evaluated against a number of best opponents (up to 30 in our study) which are taken randomly

from its adversary’s Hall of Fame. The Hall of Fame of predators is populated with the best individual of GAU after each generation avoiding repetitions. The Hall of Fame of preys follows the same policy using the best escaper from GAE.

When the maximum number of evaluations is reached, a master tournament is conducted in which predators and preys from their respective Hall of Fames are faced each other in order to obtain the best specimens of each side, becoming in this way

TABLE 3 | Optimisation results for CROMM-MS after performing master tournament between the best prey and predators obtained from the 30 independent runs of CompCGA. Additionally, results of CROMM are also provided.

# Escapers	Fitness values						Wilcoxon <i>p</i> -value
	CROMM		CROMM-MS		Improvement		
	Average	Best	Average	Best	Average	Best	
2	289.343	118.292	233.603	58.292	19.3%	50.7%	0.047
4	354.785	134.773	267.587	81.715	24.6%	39.4%	0.006
8	598.123	171.215	354.673	98.463	40.7%	42.5%	0.000
16	703.281	145.598	341.886	96.854	51.4%	33.5%	0.001
Average	486.383	142.470	299.437	83.831	38.4%	41.2%	—

Best results are in bold.

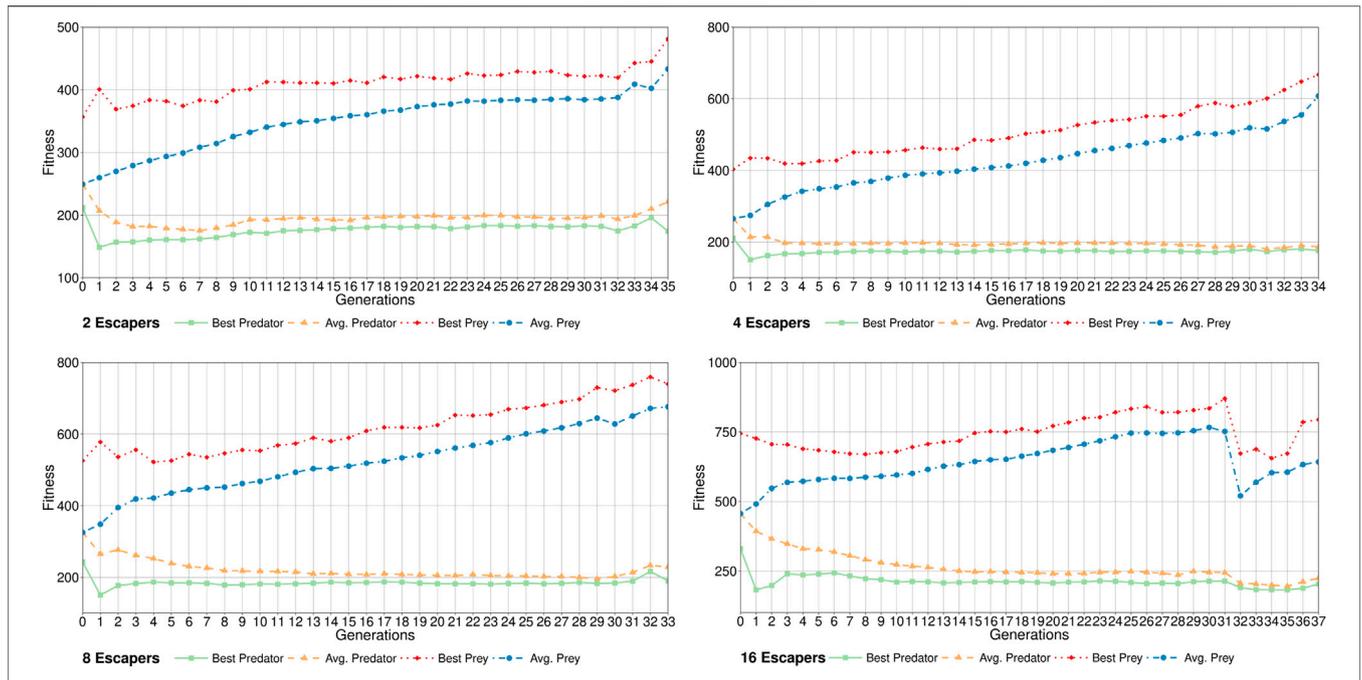


FIGURE 7 | Average convergence of Predators and Preys of the 30 runs of CompCGA for each case study.

the solutions (predator and prey configurations) calculated by each CompCGA’s run. As this is a stochastic algorithm we perform 30 independent run for each case study (described in the next Section). **Table 2** summarises the parameters of the proposed CompCGA.

3.4 Case Studies

Four case studies are analyzed in this article comprising 2, 4, 8, and 16 escapers which try to run away from a restricted area. The surveillance system is made of four swarms of 4 UMVs, four swarms of 4 UAVs, and one swarm of 2 UGVs, i.e. 34 autonomous vehicles in total.

The analyzed scenario is set up in a fictional island having a restricted area in its center (**Figure 5**). Three patrolling areas were defined, the innermost for UGVs where flying devices are not allowed for safety reasons. The central area is assigned to UAVs, which are faster and have cameras pointing toward the ground, and finally, the outer patrolling area comprises water surface so that UMVs are the last detection barrier to be defeated by escapers. There are shared borders between patrolling areas, i.e. 1-m wide regions where both type of vehicles can coexist, to easy the transition between different type of swarms. Since those vehicles are moving at different altitudes (UMV/UAV and UAV/UGV) there is no risk of collision in such shared areas.

The map dimensions are 400 × 400 m, the simulation time was set to 10 min and the escapes can take place during the time slot going from 100 to 400 s. The bottom limit is for allowing the initial spread of the swarms and the top limit is to give enough time to escapers to reach the border of the map.

3.5 Evaluation

In order to evaluate each individual representing the configuration of predators and preys we use the Hunted Sim (Stolfi et al., 2020b, Stolfi et al., 2020d). Hunted sim is a simulation environment dedicated to simulate diverse unmanned vehicles in different scenarios involving not only escapers but also intruders. It considers a map divided in 1 × 1-m cells by which vehicles move following a mobility model (**Figure 6**).

Each individual is represented by a vector \vec{x} (**Eq. 1**) containing the configuration of the $N = 34$ predators and the M preys ($M \in \{2, 4, 8, 16\}$, depending on the case study). The evaluation of \vec{x} using the simulator and the fitness function (**Eq. 2**) provides the fitness value of each individual. The fitness function averages the detection distance (δ) of each escaper spotted and adds a penalisation constant ω for each missing escaper (η). The value of ω is the maximum distance from the map center to its corners. It is calculated according to **Eq. 3** where

TABLE 4 | Results for CROMM-MS after testing each best configuration obtained in the four case studies (2, 4, 8, and 16 escapers).

# Escapers	CROMM-MS			
	2	4	8	16
2	233.6	291.2	324.5	286.9
4	294.5	267.6	285.4	318.8
8	430.7	367.7	354.7	405.6
16	605.8	588.7	637.0	341.9
Average	391.1	378.8	400.4	338.3

Best individual results and the best average value are in bold.

TABLE 5 | Metrics obtained from the surveillance results of CROMM-MS compared with CROMM. CROMM-MS was configured with the best configuration achieved for the case study with 16 escapers.

# Escapers	Coverage (%)		Detections (%)		Avg. Distance (m)	
	CROMM	CROMM-MS	CROMM	CROMM-MS	CROMM	CROMM-MS
2	82.0	89.2	83.3	83.3	195.1	192.6
4	82.0	89.2	84.2	86.7	175.7	168.0
8	82.0	89.2	81.3	89.2	173.9	160.5
16	82.0	89.2	88.3	96.0	175.3	162.8
Average	82.0	89.2	84.3	88.8	180.0	171.0

Best results are in bold.

D_x and D_y are the width and height of the map. Predators (GAU) focus on minimising the detection distance and detecting a maximum number of escapers, consequently, the lower the value of $F(\vec{x})$, the better. On the opposite side (GAE), the preys' objective is to maximise its fitness value by being detected as far as possible from the restricted area or even reaching the border of the map if possible.

$$\vec{x} = \{r_1, \dots, r_N, \epsilon_{b_1}, \epsilon_{c_1}, \epsilon_{s_1}, \epsilon_{t_1}, \epsilon_{a_1}, \dots, \epsilon_{b_M}, \epsilon_{c_M}, \epsilon_{s_M}, \epsilon_{t_M}, \epsilon_{a_M}\}, \quad (1)$$

$$F(\vec{x}) = \bar{\delta} + \omega \times \eta, \quad (2)$$

$$\omega = \frac{1}{2} \sqrt{D_x^2 + D_y^2}. \quad (3)$$

4 RESULTS

The experimentation done consisted of the optimisation of CROMM-MS using the CompCGA followed by the analysis of the results obtained including surveillance metrics for each case study, and the comparison with CROMM. Note that CROMM was not optimised since it does not present any parameter.

Thirty runs of CompGGA were performed where the parameters of each vehicle in the swarm were optimised to maximise the escaper's detection rate using the evaluation function discussed in Section 3.5. Table 3 shows the fitness of the best individual (predator specimen) after performing the master tournament between the configurations stored in both Hall of Fames (predators vs. preys). Additionally, the fitness values for CROMM were obtained to test this mobility model and know how it performs against the best escapers (preys). All in all, CROMM-MS has achieved lower (better) fitness values on average and best values than CROMM in the four case studies analyzed. CROMM-MS has achieved an average improvement of 38.4% over CROMM, and the best fitness value is 41.2% better when using our proposal in the four case studies. These results has a statistical significance greater than 95% (greater than 99% for 4, 8, and 16 escapers) which has been calculated using the Wilcoxon test.

Figure 7 shows the average evolution of fitness values for the 120 runs (30 per case study) of CompCGA. It can be seen that initially both populations have approximately the same average fitness value which is evolving to lower values (minimisation) for predators and higher values (maximisation) for preys. In the case study with 16 escapers a sharp change in this tendency is observed

in generation 31 where the GAU (predators) have found a new good configuration for vehicles hard to beat by the preys. Note that all the best individuals collected from each generation are stored in their respective Hall of Fame to be used in the final master tournament.

As the number of escapers in a real world scenario is *a priori* unknown, we have taken the best predators from the optimisation process (of each case study) and tested them in all our case studies as shown in Table 4. By doing so, we have chosen the configuration for the vehicles which is the most successful not in one but in the four different case studies proposed in our approach. The results show that although each configuration achieved the best results for its own case study, the configuration for 16 escapers turned out to be the best on average.

Based on these results we analyze the other metrics of the system using the configuration for 16 escapers. Table 5 shows these metrics where the surveillance results of CROMM-MS are compared with CROMM. It can be seen that not only the former achieved a better detection rate (88.8% vs. 84.3% on average), but also detections occurred closer to the restricted area (171 vs. 180 m on average). Furthermore, the area covered by the autonomous vehicles using CROMM-MS was greater (89.2%) than when using CROMM

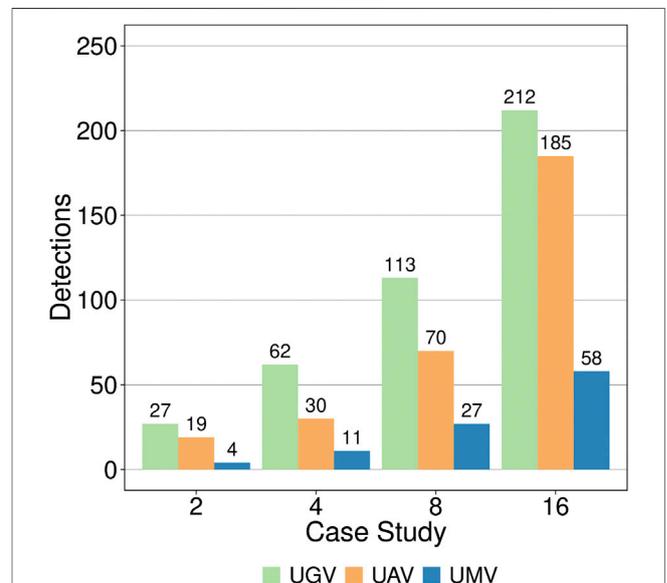


FIGURE 8 | Escaper detections grouped by case study and vehicle type.

(82%). All these results plus the statistical tests provided (in terms of fitness values), confirm that our approach using a fine parameterisation effectively improves the performance of a heterogeneous multi-swarm surveillance system.

Finally, a last study regarding the number of detections arranged by vehicle type was done. It can be seen in **Figure 8** that UGVs are spotting the majority of escapers as they are the first obstacle to be overcome and our evaluation function prioritises early detections. In the midfield UAVs are doing almost the rest of detections and the last barrier, i.e. UGVs, make the rest of detections. This last study supports the idea of using multi-swarms of vehicles and several patrolling areas as a viable surveillance system in which each vehicle's characteristics are exploited to improve the system efficiency as a whole.

5 DISCUSSION

In this paper we presented a surveillance system based on a new mobility model called CROMM-MS (Chaotic Rössler Mobility Model for Multi-Swarms) with the aim of patrolling and detecting individuals escaping from a restricted area.

We have proposed the parameterisation of CROMM (Chaotic Rössler Mobility Model) in order to address heterogeneous multi-swarms (UAVs, UGVs and UGVs) where early detection has priority over coverage. A new Competitive Coevolutionary Genetic Algorithm (CompCGA) was designed to optimise vehicles trajectories as well as escapers' evasion ability using a predator-prey approach.

The results obtained after 30 independent runs of CompCGA for four case studies (34 autonomous vehicles vs. 2, 4, 8, and 16 escapers) show that CROMM-MS has successfully detected 89% of escapers, performing better than CROMM (84%), not only in terms on early detection of escapers, but also in area coverage (89% vs. 82%).

As a matter of future work, we would like to improve our system even more using other techniques for spreading the swarm such as virtual pheromones or ghost vehicles, increasing the detection rates as well as area coverage. Moreover, we believe that CROMM-MS could be also adapted to detect intruders trespassing a restricted area. Consequently, we

would like to try our approach in this kind of scenarios as well as different map sizes, geographical characteristics as well as using different swarm members. Despite the competitive results achieved by CompCGA, we would like to test different optimisation approaches for our surveillance system, e.g. Differential Evolution (DE) and Particle Swarm Optimisation (PSO).

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

Conceptualization, DS; methodology, DS; software, DS; validation, DS and GD; formal analysis, DS; writing—original draft preparation, DS; writing—review and editing, DS, MB and GD; visualization, DS; supervision, GD and PB; project administration, GD and PB; funding acquisition, GD and PB.

FUNDING

This research was partially funded by the Office of Naval Research, grant number N62909-18-1-2176. The joint research program UL/SnT-ILNAS on Digital Trust for Smart-ICT

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 program 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 (Varrette et al., 2014) – see <https://hpc.uni.lu>.

REFERENCES

- Arbanas, B., Ivanovic, A., Car, M., Orsag, M., Petrovic, T., and Bogdan, S. (2018). Decentralized planning and control for UAV-UGV cooperative teams. *Aut. Robots* 42, 1601–1618. doi:10.1007/s10514-018-9712-y
- Batista da Silva, L. C., Bernardo, R. M., de Oliveira, H. A., and Rosa, P. F. F. (2017). Multi-UAV agent-based coordination for persistent surveillance with dynamic priorities. in Proceedings of the 2017 International Conference on Military Technologies (ICMT), Brno, Czech Republic, May 31–June 2, 2017, IEEE, 765–771. doi:10.1109/MILTECHS.2017.7988859
- Boussaa, M., Kessentini, W., Kessentini, M., Bechikh, S., and Ben Chikha, S. (2013). "Competitive coevolutionary code-smells detection." in International symposium on search based software engineering. Berlin, Germany: Springer. 50–65. doi:10.1007/978-3-642-39742-4_6
- Brust, M. R., Danoy, G., Bouvry, P., Gashi, D., Pathak, H., and Goncalves, M. P. (2017). Defending against intrusion of malicious UAVs with networked UAV defense swarms. in Proceedings of the 2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops), Singapore, October 9, 2017. IEEE, 103–111. doi:10.1109/LCN.Workshops.2017.71
- Brust, M. R., and Strimbu, B. M. (2015). A networked swarm model for UAV deployment in the assessment of forest environments. In 2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, April 7–9, 2015, IEEE, 1–6. doi:10.1109/ISSNIP.2015.7106967
- Costanzi, R., Fenucci, D., Manzari, V., Micheli, M., Morlando, L., Terracciano, D., et al. (2020). Interoperability among unmanned maritime vehicles: review and first in-field experimentation. *Front. Robot. AI* 7. doi:10.3389/frobt.2020.00091
- Deka, A., and Sycara, K. (2020). Natural emergence of heterogeneous strategies in artificially intelligent competitive teams. Available at: <https://arxiv.org/abs/2007.03102>.
- Girma, A., Bahadori, N., Sarkar, M., Tadevos, T. G., Behnia, M. R., Mahmoud, M. N., et al. (2020). IoT-enabled autonomous system collaboration for disaster-area management. *IEEE/CAA J. Autom. Sinica* 1, 1–14. doi:10.1109/JAS.2020.1003291

- Goldberg, D. E., and Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms* 1, 69–93. doi:10.11.101.949410.1016/b978-0-08-050684-5.50008-2
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. 1st edn. Boston, MA: Addison-Wesley Longman Publishing Co., Inc.
- Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. Cambridge, MA: MIT Press.
- Hood, S., Benson, K., Hamod, P., Madison, D., O’Kane, J. M., and Rekleitis, I. (2017). Bird’s eye view: cooperative exploration by UGV and UAV. In 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, June 13–16, 2017: IEEE, 247–255. doi:10.1109/ICUAS.2017.7991513
- Huang, L., Zhou, M., Hao, K., and Hou, E. (2019). A survey of multi-robot regular and adversarial patrolling. *IEEE/CAA J. Autom. Sinica* 6, 894–903. doi:10.1109/JAS.2019.1911537
- Iba, T., and Shimonishi, K. (2011). The origin of diversity: thinking with chaotic walk. *Unifying Themes in Complex Systems* 8, 447–461.
- Jayavelu, S., Kandath, H., and Sundaram, S. (2018). Dynamic area coverage for multi-UAV using distributed UGVs: a two-stage density estimation approach. In 2018 Second IEEE International Conference on Robotic Computing (IRC), Laguna Hills, CA, January 31–February 2, 2018, IEEE, 165–166. doi:10.1109/IRC.2018.00033
- Jeong, H. Y., Song, B. D., and Lee, S. (2019). Truck-drone hybrid delivery routing: payload-energy dependency and No-Fly zones. *Int. J. Prod. Econ.* 214, 220–233. doi:10.1016/j.ijpe.2019.01.010
- Lee, S., Jain, S., Yuan, Y., Zhang, Y., Yang, H., Liu, J., et al. (2019). Design and development of a DDDAMS-based border surveillance system via UVs and hybrid simulations. *Expert Syst. Appl.* 128, 109–123. doi:10.1016/j.eswa.2019.03.034
- Makkapati, V. R., and Tsiotras, P. (2020). Apollonius allocation algorithm for heterogeneous pursuers to capture multiple evaders. Available at: <https://arxiv.org/abs/2006.10253>.
- Makkapati, V. R., and Tsiotras, P. (2019). Optimal evading strategies and task allocation in multi-player pursuit-evasion problems. *Dyn Games Appl.* 9, 1168–1187. doi:10.1007/s13235-019-00319-x
- McNeal, G. S. (2016). Drones and the future of aerial surveillance. *George Washington Law Review Arguing* 84, 354–416.
- Moysis, L., Petavratzis, E., Volos, C., Nistazakis, H., and Stouboulos, I. (2020). A chaotic path planning generator based on logistic map and modulo tactics. *Robot. Auton. Syst.* 124, 103377. doi:10.1016/j.robot.2019.103377
- Petavratzis, E. K., Volos, C. K., Moysis, L., Stouboulos, I. N., Nistazakis, H. E., Tombras, G. S., et al. (2019). An inverse pheromone approach in a chaotic mobile robot’s path planning based on a modified logistic map. *Technologies* 7, 84. doi:10.3390/technologies7040084
- Pierson, A., Wang, Z., and Schwager, M. (2017). Intercepting rogue robots: an algorithm for capturing multiple evaders with multiple pursuers. *IEEE Robot. Autom. Lett.* 2, 530–537. doi:10.1109/LRA.2016.2645516
- Portugal, D., and Rocha, R. P. (2016). Cooperative multi-robot patrol with Bayesian learning. *Aut. Robots* 40, 929–953. doi:10.1007/s10514-015-9503-7
- Rizk, Y., Awad, M., and Tunstel, E. W. (2019). Cooperative heterogeneous multi-robot systems. *ACM Comput. Surv.* 52, 1–31. doi:10.1145/3303848
- Rosalie, M., Danoy, G., Chaumette, S., and Bouvry, P. (2018). Chaos-enhanced mobility models for multilevel swarms of UAVs. *Swarm and Evolutionary Computation* 41, 36–48. doi:10.1016/j.swevo.2018.01.002
- Rössler, O. E. (1976). An equation for continuous chaos. *Phys. Lett.* 57, 397–398. doi:10.1016/0375-9601(76)90101-8
- Stolfi, D. H., Brust, M. R., Danoy, G., and Bouvry, P. (2020c). Emerging inter-swarm collaboration for surveillance using pheromones and evolutionary techniques. *Sensors* 20. doi:10.3390/s20092566
- Stolfi, D. H., Brust, M. R., Danoy, G., and Bouvry, P. (2020a). A cooperative coevolutionary approach to maximise surveillance coverage of UAV swarms. In 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, January 10–13, 2020, IEEE, 1–6. doi:10.1109/ccnc46108.2020.9045643
- Stolfi, D. H., Brust, M. R., Danoy, G., and Bouvry, P. (2020b). Competitive evolution of a UAV swarm for improving intruder detection rates. In 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), New Orleans, LA, May 18–22, 2020, IEEE, 528–535. doi:10.1109/IPDPSW50202.2020.00094
- Stolfi, D. H., Brust, M. R., Danoy, G., and Bouvry, P. (2020d). “Optimizing the performance of an unpredictable UAV swarm for intruder detection,” in *Optimization and learning*. Editors B. Dorransoro, P. Ruiz, J. C. de la Torre, D. Urda, and E.-G. Talbi (Cham, Switzerland: Springer International Publishing), 37–48. doi:10.1007/978-3-030-41913-4_4
- Tiguercha, A., Ladjici, A. A., and Boudour, M. (2014). Competitive co-evolutionary approach to stochastic modeling in deregulated electricity market. In Proceedings of the 2014 IEEE International Energy Conference (ENERGYCON), Cavtat, Croatia, May 13–16, 2014, IEEE, 514–519. doi:10.1109/ENERGYCON.2014.6850475
- Varrette, S., Bouvry, P., Cartiaux, H., and Georgatos, F. (2014). Management of an academic HPC cluster: the UL experience. In Proceedings of the 2014 International Conference on High Performance Computing & Simulation (HPCS), Bologna, Italy, July 21–25, 2014, IEEE, 959–967. doi:10.1109/HPCSim.2014.6903792
- Volos, C. K., Kyprianidis, I. M., and Stouboulos, I. N. (2012). A chaotic path planning generator for autonomous mobile robots. *Robot. Auton. Syst.* 60, 651–656. doi:10.1016/j.robot.2012.01.001
- Vu, Q., Raković, M., Delic, V., and Ronzhin, A. (2018). “Trends in development of UAV-UGV cooperation approaches in precision agriculture,” in *Interactive collaborative robotics*. Editors A. Ronzhin, G. Rigoll, and R. Meshcheryakov (Cham, Switzerland: Springer International Publishing), 213–221. doi:10.1007/978-3-319-99582-3_22
- Waslander, S. L. (2013). *Unmanned aerial and ground vehicle teams: recent work and open problems*. Berlin, Germany: Springer, 21–36. doi:10.1007/978-4-431-54276-6_2
- Wiegand, R. P., Liles, W. C., and Jong, K. A. D. (2001). An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. In Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation GECCO’01, San Francisco, CA, July 2001, (Burlington, MA: Morgan Kaufmann Publishers Inc.), 1235–1242.
- Yan, R.-j., Pang, S., Sun, H.-b., and Pang, Y.-j. (2010). Development and missions of unmanned surface vehicle. *J. Mar. Sci. Appl.* 9, 451–457. doi:10.1007/s11804-010-1033-2

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Stolfi, Brust, Danoy and Bouvry. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.