







## Article

# Coverage Strategy for Target Location in Marine Environments Using Fixed-Wing UAVs

Javier Muñoz \* , Blanca López , Fernando Quevedo , Concepción A. Monje , Santiago Garrido   
and Luis E. Moreno 

Robotics Lab, Universidad Carlos III de Madrid, Av. Madrid 30, 28911 Leganés, Spain; bllopezp@ing.uc3m.es (B.L.); fquevedo@ing.uc3m.es (F.Q.); cmonje@ing.uc3m.es (C.A.M.); sgarrido@ing.uc3m.es (S.G.); moreno@ing.uc3m.es (L.E.M.)

\* Correspondence: jamunozm@ing.uc3m.es

**Abstract:** In this paper, we propose a coverage method for the search of lost target or debris on the ocean surface. The OSCAR data set is used to determine the marine currents and the differential evolution genetic filter is used to optimize the sweep direction of the lawnmower coverage and get the sweep angle for the maximum probability of containment. The position of the target is determined by a particle filter, where the particles are moved by the ocean currents and the final probabilistic distribution is obtained by fitting the particle positions to a Gaussian probability distribution. The differential evolution algorithm is then used to optimize the sweep direction that covers the highest probability of containment cells before the less probable ones. The algorithm is tested with a variety of parameters of the differential evolution algorithm and compared to other popular optimization algorithms.

**Keywords:** coverage path planning; UAVs; genetic algorithms; differential evolution; search algorithms



**Citation:** Muñoz, J.; López, B.; Quevedo, F.; Monje, C.A.; Garrido, S.; Moreno, L.E. Coverage Strategy for Target Location in Marine Environments Using Fixed-Wing UAVs. *Drones* **2021**, *5*, 120. <https://doi.org/10.3390/drones5040120>

Academic Editor: Diego González-Aguilera

Received: 30 August 2021

Accepted: 13 October 2021

Published: 17 October 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Searching missions started to gain value in the maritime field during World War II, when the US Army tried to detect enemy submarines. Currently, these exploration tasks have received a great amount of attention for different reasons, focused on civilian applications. Current efforts target search and rescue missions of passengers affected by a maritime accident and the search for wreckage scattered across the ocean. Search operations on the ocean surface remain a complex task due to the dynamics of the wind and ocean currents, spreading lost targets and debris in an unpredictable way. A remarkable real scenario that inspired this field of research is the search for Malaysian Flight 370 (MH370). The aircraft disappeared in the middle of the ocean midflight in March 2014, after takeoff from Kuala Lumpur on its way to Beijing carrying 239 people. The search operations covered an area of over 4,600,000 km<sup>2</sup> in the Indian Ocean and involved a total of 40 ships and aircrafts. The search did not yield any evidence and was consequently canceled [1].

Numerous researchers have used the MH370 case to develop algorithms capable of tracking and finding lost debris from wrecks. Miron [2] et al. used a method based on Markov chains. They developed a framework that uses the locations and times of confirmed airplane debris beachings and historical trajectories produced by drifters to restrict the location of the crash site. García-Garrido [3] et al. proposed the use of Lagrangian descriptors in combination with publicly available ocean data sources to assess the spatiotemporal state of the ocean in the priority search area at the time of impact and the following weeks. Trinanés [4] et al. also used data sets of surface drifters to backtrack ocean debris based on the location where it was found that provides information on the potential crash site.

Some more recent approaches such as that of [5] use a team of UAVs with MIMO radars to estimate the direction of arrival of the target ships by reconstructing the support of a block sparse matrix. The disadvantage is that a ship with a GPS signal needs to be deployed

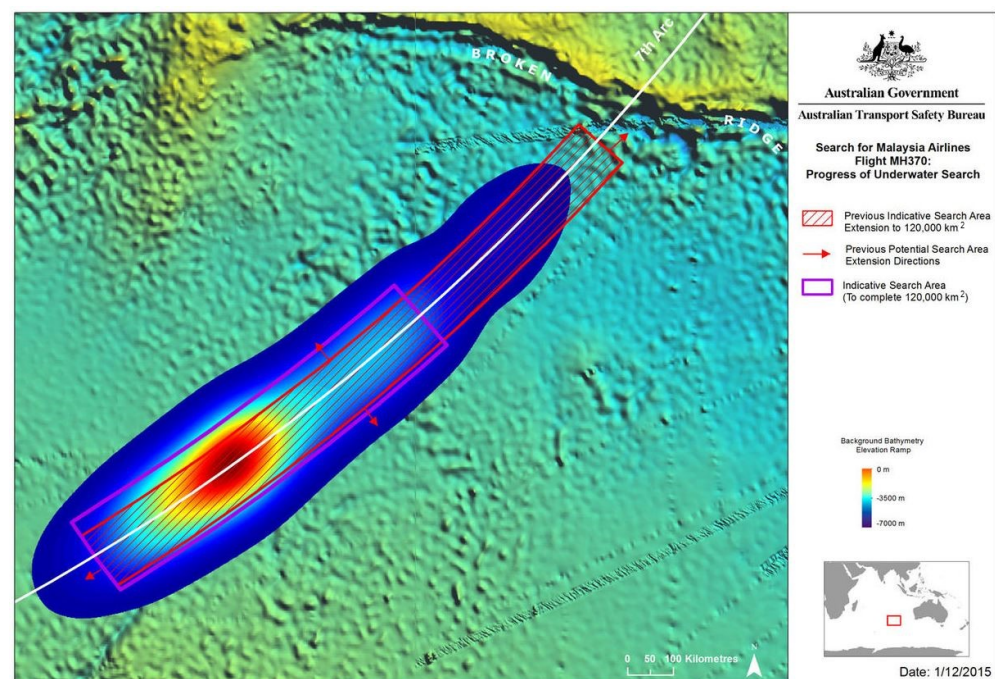
to know the UAVs and targets positions. The approach proposed by Xiao et al. [6] uses UAVs to help in marine mass casualty incident search and rescue by providing additional information to an USV operator. This method requires one operator for the USV and another one for the UAV.

Several approaches treat the search problem from different scopes, such as computer vision [7,8], wireless sensor network design [9,10], task allocation formulations [11,12], iterative optimization algorithms [13], and optimal path planning calculations for autonomous vehicles to help in the search [14,15]. In this work, we aim at the latter, proposing an algorithm capable of providing the route for the vehicles to follow.

There are two main characteristics that define the design process of a search mission, mainly concerning the maritime field application [16]:

- Definition of the target area. Since the weather conditions in maritime environments are usually inaccurate or unknown and the air and ocean currents can drift the debris away from their last known position, it is necessary to establish a method to define the search target area.
- Design of a search path. Even if the target area is correctly defined, there is no optimal solution for the design of the search trajectory problem. In this work, the goal is to compute paths along the whole searching area that cover the points with the higher probability of containment as fast as possible.

Regarding the establishment of the target area, the probability of containment is the first factor to be determined. This measure is defined as the probability to find the target in each specific region of the search area. Following this methodology, the search area is decomposed into cells and then turned into a probability map by assigning these partitions the probability value related to the chances of finding the search target. The resulting map will represent the possible positions of the targets, considering the ocean and wind currents. An example can be seen in Figure 1, in which zones with lower and higher probability of containment indices are distinguished through a color code. From that target position, a probability of containment map is calculated by deploying drifter particles around the target that follow the marine and wind currents and fitting them to a Gaussian probability distribution.



**Figure 1.** Malaysian Airlines MH370 location probability heat map.

In this paper, we propose a target position estimation method and a coverage path planning algorithm for fixed-wing UAVs based on drifter particles and the differential

evolution method. The proposed method has a great amount of potential in real world applications since the movement of the particles representing the possible positions of the lost target is based on real world current data from the OSCAR data set. Furthermore, a large number of particles is used to ensure a good representation of the lost target movement, as well as some noise applied to the OSCAR data to generate more possible outcomes for the drifter particles.

This work is aimed for fixed-wing UAVs based on the fact that they achieve greater speeds and have a greater autonomy, which is of great importance when tackling a marine search and rescue problem. They are also more wind-resistant, can carry heavier payloads, which can help in the search and rescue missions, have greater stability, and can recover more easily from sudden power loss, which is important in marine environments. Moreover, they can achieve greater altitudes; when they are equipped with a long-range radar or camera, they can greatly improve the chances of target location. Fixed-wing UAVs are also better than multirotor UAVs when performing linear flights that, when considering the dimensions of the map and the length of each band, are going to be applied to our particular problem.

The use of UAVs is key in situations such as the one presented in this work, where there has been an accident or the target is missing, and it is necessary to perform a fast search. UAVs have a great advantage over USVs, since they have a better line of sight and can detect targets from afar. Furthermore, the depth perception from a USV is difficult, since objects may be far apart even though they appear close, and it is an even harder task when there are high and strong waves. These complications disappear when using UAVs due to the fact that they have a clear top view from the sky and their depth perception is not affected. Since the results in this work focus on obtaining the coverage path and no real UAVs have been used, no UAV-specific characteristics have been taken into account. However, the coverage bandwidth and the desired height of the coverage path can be defined, making the algorithm able to adapt to any fixed-wing UAV.

Note that in the context of real world applications, UAVs need a ground base station and a piece of terrain to take off and land. Furthermore, the ground station needs to be within range of the target area for the coverage mission such that the UAV meets the necessary energy constraints to perform the mission and come back safely. Some applications use an aircraft carrier to carry the UAVs and act as the base station.

The differential evolution method is used to optimize the coverage angle for the search task, based on the probability of containment (POC) of each cell of the probability map, created from a Gaussian probability distribution generated from the positions of the drifter particles. Global optimization algorithms are necessary in fields such as engineering, statistics, and finance. However, many of these practical problems have objective functions that are nondifferentiable, noncontinuous, noisy, multidimensional, or have many local minima and constraints. These problems are difficult to solve analytically, so algorithms such as differential evolution can be used to find approximate solutions. The DE algorithm is the one used in this work since it has proved useful in previous publications, but any optimization algorithm with similar characteristics could be used.

The rest of the paper is organized as follows. Section II describes the methodology used for the target position estimation and introduces the coverage path planning method. Section III presents the results of the simulations performed to test the algorithm. Finally, in section IV, the main conclusions of the work are outlined.

## 2. Materials and Methods

This section details both the target position estimation of the target method and the coverage path planning algorithm used to search for the target. The target estimation of the target position is performed by dropping drifter particles that move with the ocean and wind currents and then fitting them to a Gaussian probability distribution to obtain the probability of containment map. The coverage path planning algorithm uses the differential

evolution method to obtain the best coverage angle to cover the maximum probability of containment possible and then performs back-and-forth motions to create a coverage path.

### 2.1. Target Position Estimation

For the target position estimation process, the Ocean Surface Current Analysis (OSCAR) database [17] is used to obtain an ocean current map. Once the map of the particular section of the wreckage is obtained from the database, a number of drifters are dropped around the target's last known position within a determined radius. This method of drifters tracking to estimate the direction of ocean currents has been used for the generation of data sets such as [18].

Once the drifters are placed on the map, they move following the currents given by the OSCAR database and some noise implemented due to variations in ocean drift and wind currents. The equation for calculating the next position of the particles is shown in Equation (1):

$$(x, y)^{t+1} = (x, y)^t + \Delta t((u, v) + \max E(u, v) * \text{rand}) \quad (1)$$

where  $(x, y)^{t+1}$  is the next estimated position of the particles,  $(x, y)^t$  is the current estimated position of the particles,  $(u, v)$  is the velocity vector for the  $x$  and  $y$  axes,  $\max E$  is the maximum possible error in the ocean current velocity vector, and  $\text{rand}$  is a random number in the interval  $[0, 1]$ . In this case,  $\max E$  is calculated as  $0.1 * \text{mean}(u, v)$ .

When the search is about to start, the particle distribution is fitted to a Gaussian mixture model [19], which allows to gather all possible observations under a group of normal distributions of different parameters, as shown in Figure 2.

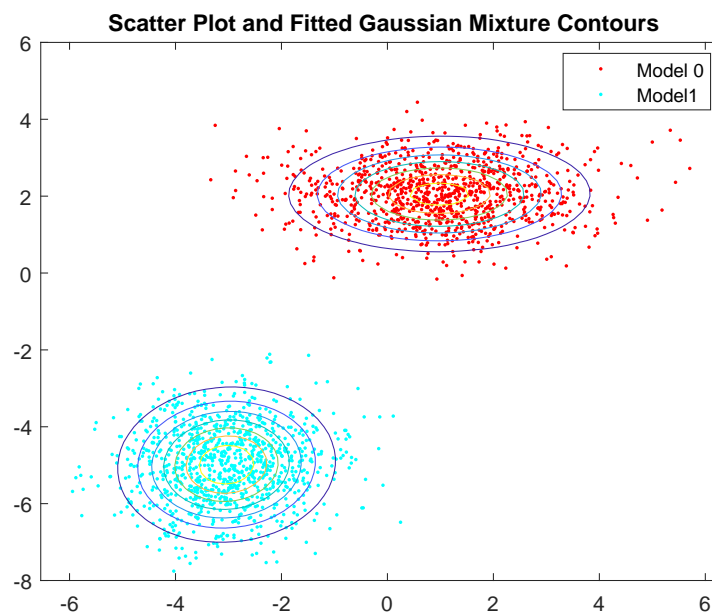
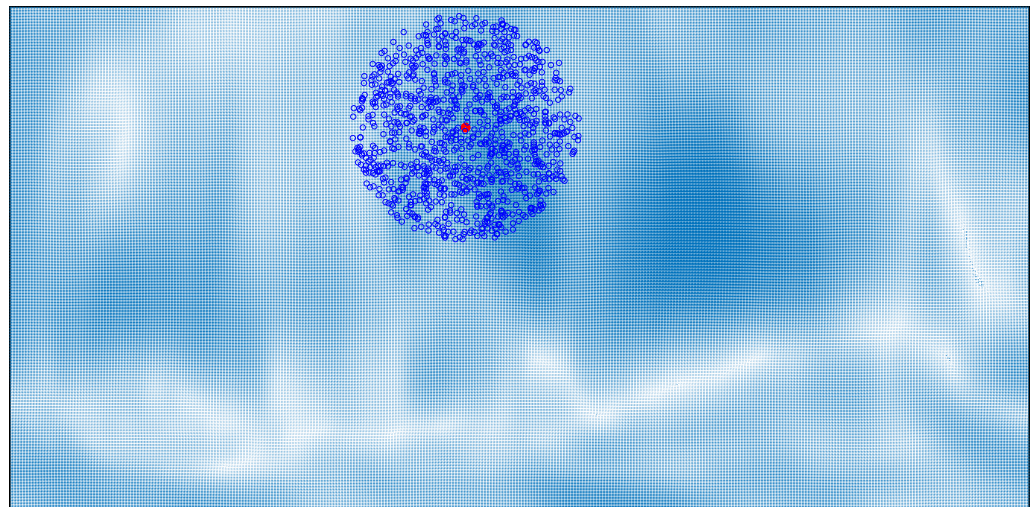


Figure 2. Data fitted to Gaussian mixture model [20].

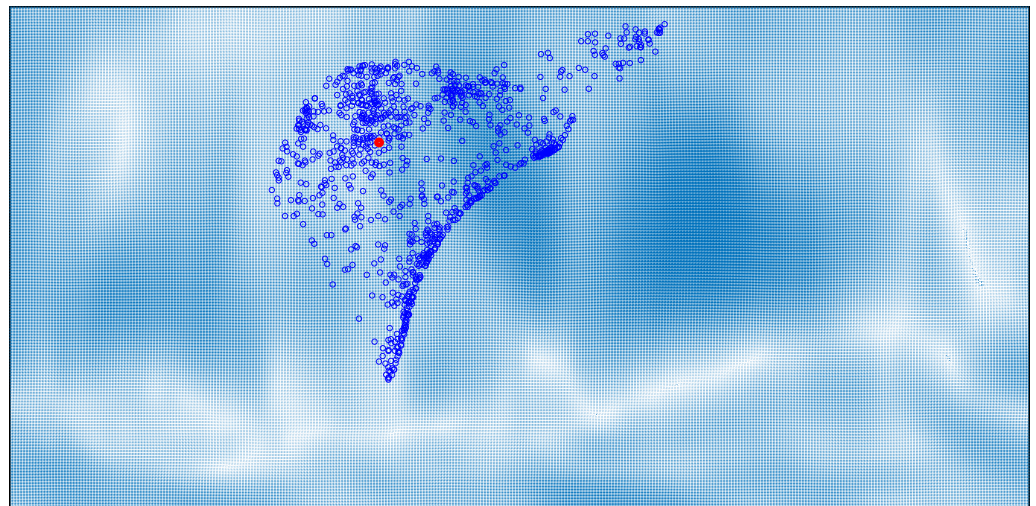
The first step is to drop the drifter particles around the target's last known position. An example of initial drifter particles distribution is shown in Figure 3. The red dot indicates the target's last known position, and the blue dots are the drifter particles scattered around that position.

Once the drifters are distributed in a circular area around the target's last known position, they start moving with the ocean and wind drift currents for the predefined duration, following Equation (1). After the desired number of iterations have passed, the drifter particles positions are saved. The final positions of the drifter particles for the currents and particle distribution used above are shown in Figure 4.





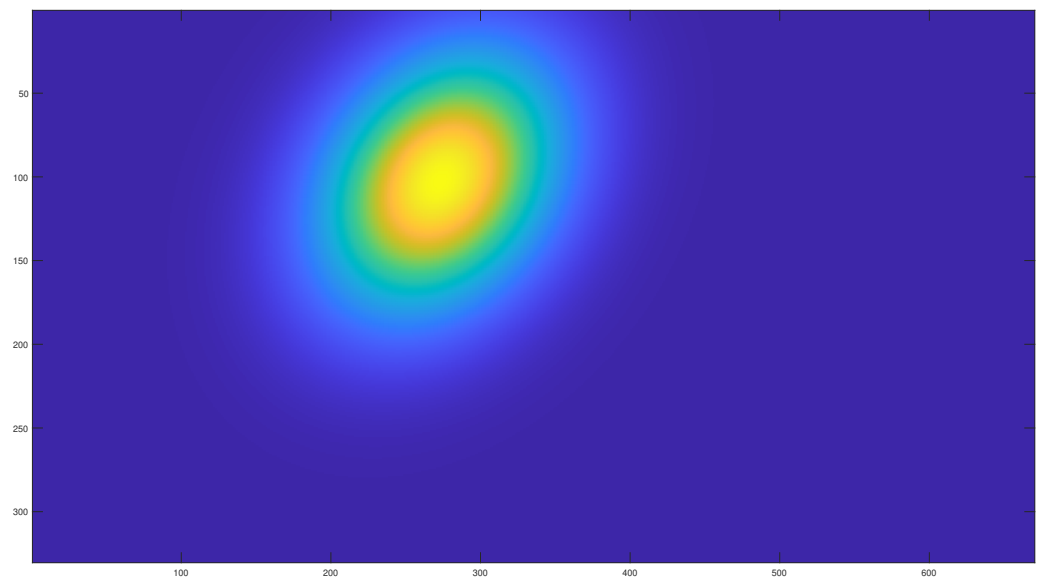
**Figure 3.** Initial position of drifter particles.



**Figure 4.** Final position of drifter particles.

When the final positions of the drifter particles are saved, they are fitted to a Gaussian probability distribution, creating the probability of containment map. This probability of containment map obtained from the position of the drifter particles can be seen in Figure 5. The cells with the highest probability of containment are shown in yellow, while the cells with low to zero probability of containment are shown in dark blue.

After the target position estimation process has been described, the following subsection focuses on obtaining the coverage path to maximize the probability of containment covered by the fixed-wing UAV.



**Figure 5.** Probability of containment map obtained by fitting the drifter particle positions to a Gaussian distribution.

## 2.2. Coverage Path Planning Algorithm

The coverage path planning algorithm used in this work is based on the differential evolution method. The differential evolution method is an evolutive filter designed to solve global optimization problems over continuous spaces, proposed by Storn and Price [21].

The evolutive filter uses a direct parallel search method that uses parameter vectors of  $n$  dimensions  $x_i^k = (x_{i,1}^k, \dots, x_{i,n}^k)^T$  to point each candidate solution  $i$  to the optimization problem in each iteration step  $k$ . This method uses  $N$  parameter vectors  $\{x_i^k; i = 0, 1, \dots, N\}$  as a population for every generation  $t$  of the optimization process.

The initial population is chosen arbitrarily to uniformly cover the whole parameter space. In the absence of a priori information, the whole parameter space has the same probability to contain the optimal parameter vector and a uniform probability distribution is assumed.

The evolutive filter generates new parameter vectors by adding the weighted difference vector between two population members to a third member, as it can be seen in Figure 6. If the resulting vector yields a lower objective function value than the corresponding member of the previous population, the newly generated vector replaces the one which with it was compared; otherwise, the old vector is conserved.

This basic idea is extended by perturbing an existing vector through the addition of one or more weighed difference vectors. This perturbation scheme generates a variation  $v$  according to the following equation:

$$v = x_i^k + L(x_b^k - x_i^k) + F(x_{r2}^k - x_{r3}^k)$$

where  $x_i^k$  is the parameter vector to be perturbed in iteration  $k$ ,  $x_b^k$  is the best parameter vector of the population in iteration  $k$ , and  $x_{r2}^k$  and  $x_{r3}^k$  are the parameter vectors randomly chosen from the rest of the population.  $L$  and  $F$  are real and constant factors that control the amplification of the differential variables  $(x_b^k - x_i^k)$  and  $(x_{r2}^k - x_{r3}^k)$ .

With the purpose of incrementing the diversity of the new parameter vector generation, the crossover concept is introduced. We denote the new parameter vector by  $u_i^k = (u_{i,1}^k, u_{i,2}^k, \dots, u_{i,n}^k)^T$  with

$$u_{i,j}^k = \begin{cases} v_{i,j}^k & \text{if } p_{i,j}^k < \delta \\ x_{i,j}^k & \text{if } p_{i,j}^k \geq \delta \end{cases}$$

where  $v_{i,j}^k$  is the exchanged parameter in member  $u_{i,j}^k$ ,  $p_{i,j}^k$  is a random value in the interval  $[0, 1]$  for each parameter  $j$  of the population member  $i$  in step  $k$ , and  $\delta$  is the crossover probability and constitutes the crossover control variable. Random values  $p_{i,j}^k$  are created for each trial vector  $i$ .

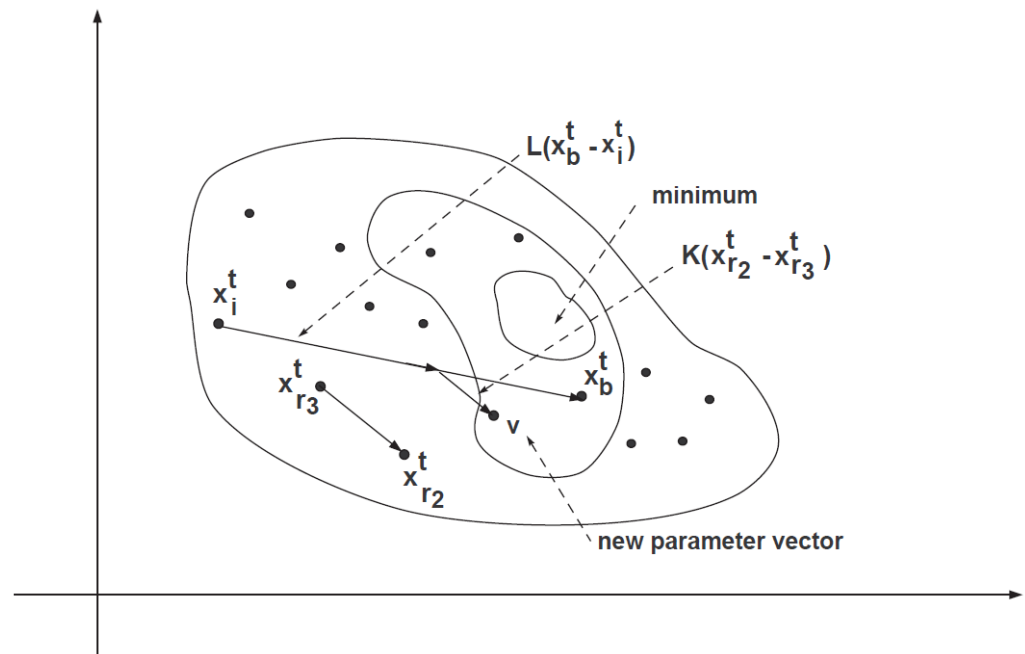


Figure 6. Generation of a new population member.

To decide whether or not vector  $u_i^k$  should become a member of generation  $i + 1$ , the new vector is compared to  $x_i^k$ . If vector  $u_i^k$  yields a better value of the objective function than  $x_i^k$ , then  $x_i^k$  is replaced by  $u_i^{k+1}$ ; on the contrary, the old value  $x_i^k$  is saved for the next generation. Mutation, crossover, and selection mechanisms are well known and can be found in the literature.

In order to calculate the coverage path, a binary mask is created by thresholding the POC (probability of containment) map. All cells with a POC value higher than a certain limit are considered into the coverage area. An example of a coverage binary mask is shown in Figure 7. Then, the initial population is created with angles ranging from  $-90$  to  $90$  degrees. For every member of the population, the coverage path is calculated with the given angle.

To test these angles, coverage bands are drawn perpendicular to the coverage angle direction. Then, all the bands that do not have enough points inside the coverage area are deleted. Afterward, the bands inside the coverage area are cut so that they keep a distance equal to the coverage bandwidth from the borders of the coverage area. Once the DE optimization process is finished and the final coverage angle is obtained, the path is calculated as explained before, but this time, the bands are linked with arcs to create a feasible path.

The cost function used with the differential evolution method to calculate the optimal coverage angle is shown in Equation (2). All calculated paths are downsampled to 1000 waypoints, so the comparison between paths is fair. Furthermore, the cost function is applied twice to every path, starting from the first and last waypoints:

$$cost = \sum_{i=1}^{1000} \frac{i}{pocMap(path(i))} \quad (2)$$

where  $pocMap$  is the probability of containment cell map,  $i$  is the path waypoints index, and  $path$  is the path calculated given a certain coverage angle.



Figure 7. Binary mask used for the coverage path planning algorithm.

To better show the process followed by the proposed algorithm, a flowchart is shown in Figure 8.

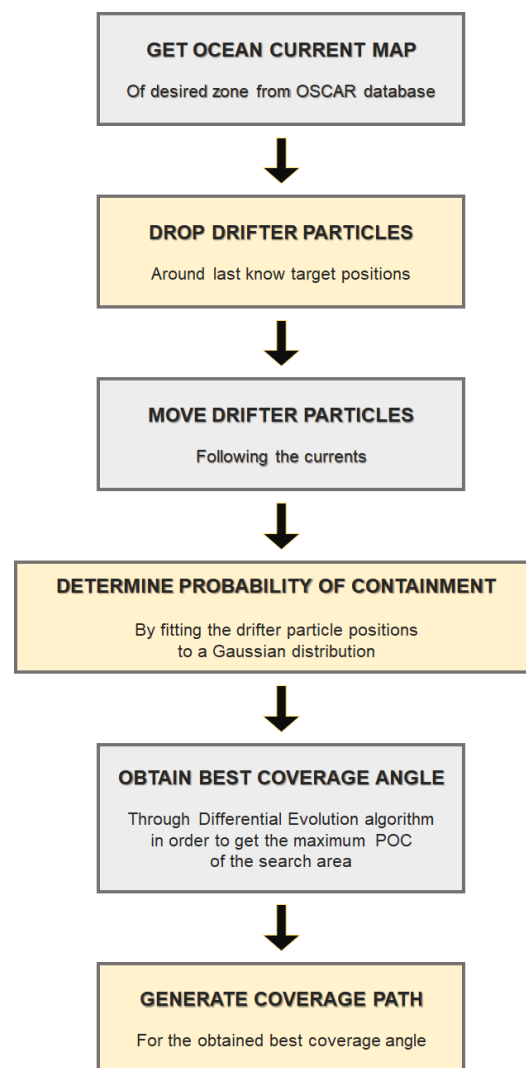


Figure 8. Schematic representation of the proposed algorithm.



### 3. Results

Since we did not have access to a real fixed-wing UAV nor did we meet the necessary regulations to operate one, the tests were performed in a simulated environment in MATLAB. To test the effectiveness of the proposed algorithm, the coverage path planning simulation was run for a different number of parameters of the DE algorithm. The machine used to run the test was a computer with 16GB RAM and an AMD Ryzen 5 5600X CPU @4.12 GHz. The first test was run for a fixed number of five members of the population and a number of iterations varying from 5 to 50. The measures collected from the tests were the total probability of containment based on the position of the waypoints of the obtained coverage path over the probability of containment distribution map, the total execution time taken by the algorithm to generate the coverage path, and the final angle obtained from the differential evolution algorithm when the iterations finished. Since the differential evolution algorithm has a stochastic nature, the measures for both the mean and the standard deviation (std) were collected. Results from this test can be seen in Table 1:

**Table 1.** Simulation results for an incrementing number of iterations of the DE algorithm.

		Number of Iterations				
		5	10	15	25	50
Time	Mean	0.2301	0.4221	0.6157	0.9981	1.8952
	Std	0.013	0.0157	0.0197	0.0400	0.0295
$\text{POC} \times 10^5$	Mean	5.0197	5.1536	5.2459	5.3407	5.4704
	Std	0.2767	0.2267	0.2647	0.2456	0.3204
Angle	Mean	10.33	12.83	11.16	14.22	16.97
	Std	46.75	46.35	48.55	48.28	47.72

The second test was run for a fixed number of 5 iterations, and a number of members of the population ranging from 5 to 50. Results from this test can be seen in Table 2:

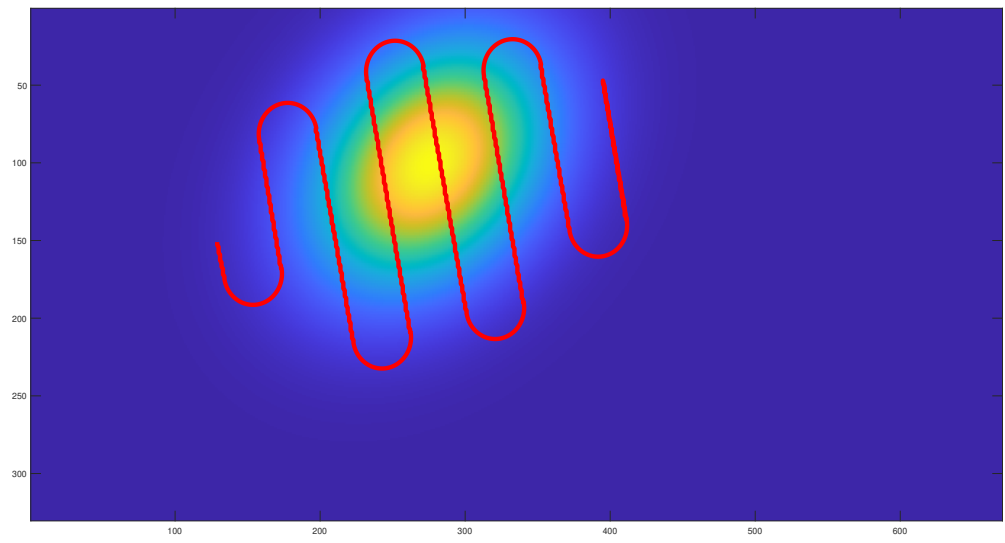
**Table 2.** Simulation results for an incrementing number of members of the population of the DE algorithm.

		Number of Population Members				
		5	10	15	25	50
Time	Mean	0.2301	0.446	0.6705	1.1309	2.2447
	Std	0.013	0.0065	0.0103	0.0504	0.0145
$\text{POC} \times 10^5$	Mean	5.0197	5.0171	5.0209	5.0343	5.036
	Std	0.2767	0.2919	0.2878	0.3	0.2725
Angle	Mean	10.33	12.62	11.25	11.39	11.53
	Std	46.75	46.97	47.47	46.07	48.23

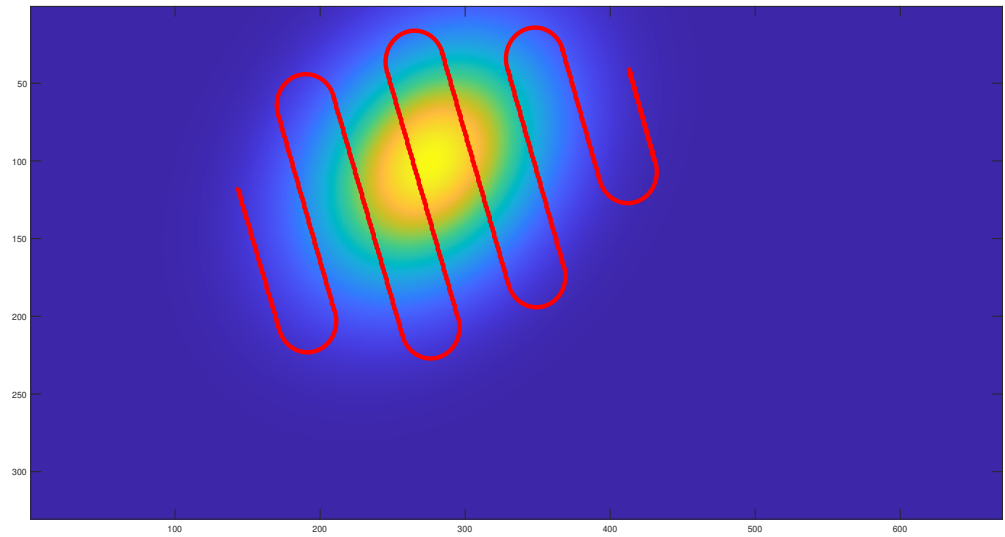
Figures 9–11 show the paths obtained from the DE algorithm with a different number of parameters.

The tables show that after a determined number of iterations and members of the population, the obtained angle converges to a similar value, which we can consider as the near optimal coverage angle. This result is better reflected in Figures 9–11, where the initial and final paths obtained are similar.

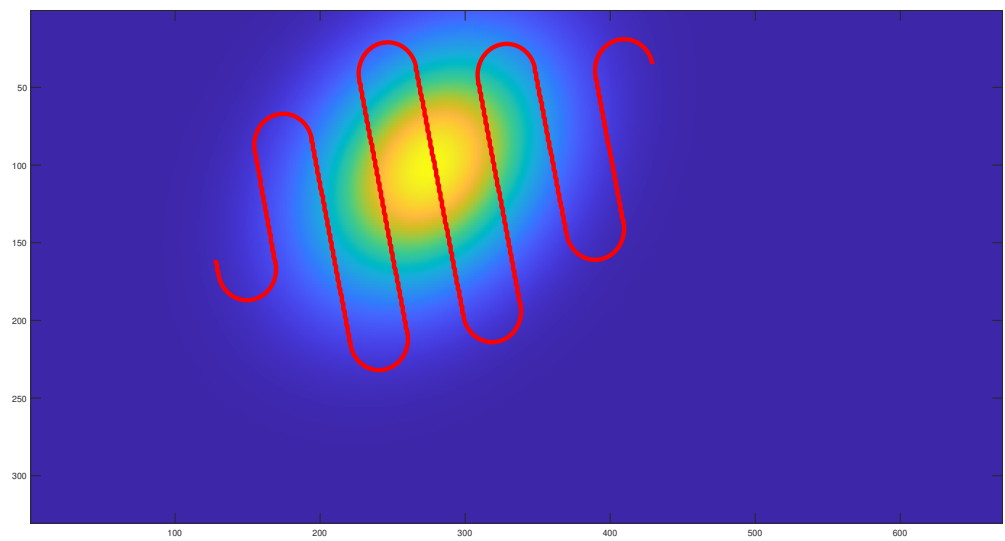




**Figure 9.** Path obtained with 5 members of the population and 5 iterations.

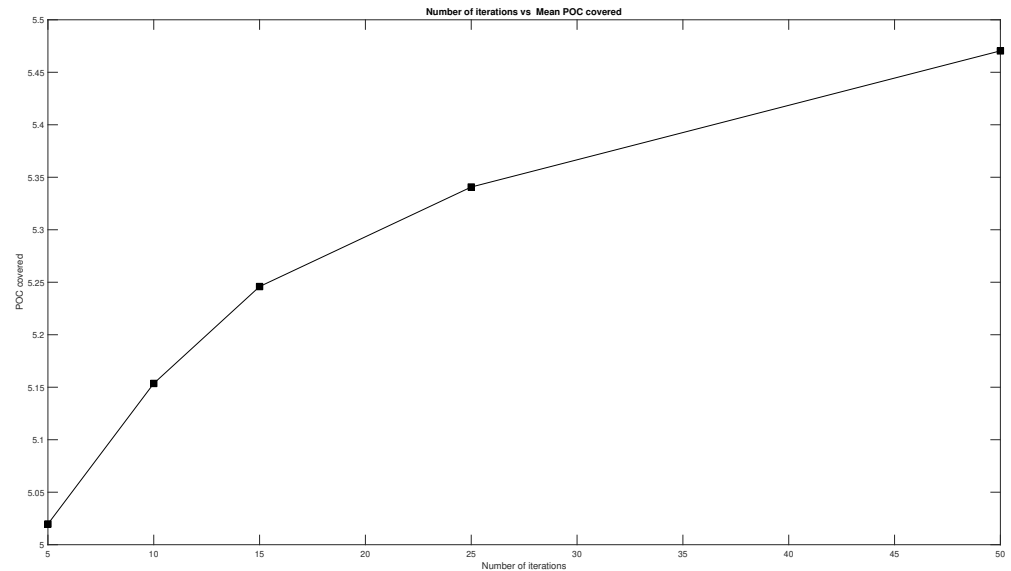


**Figure 10.** Path obtained with 5 members of the population and 50 iterations.

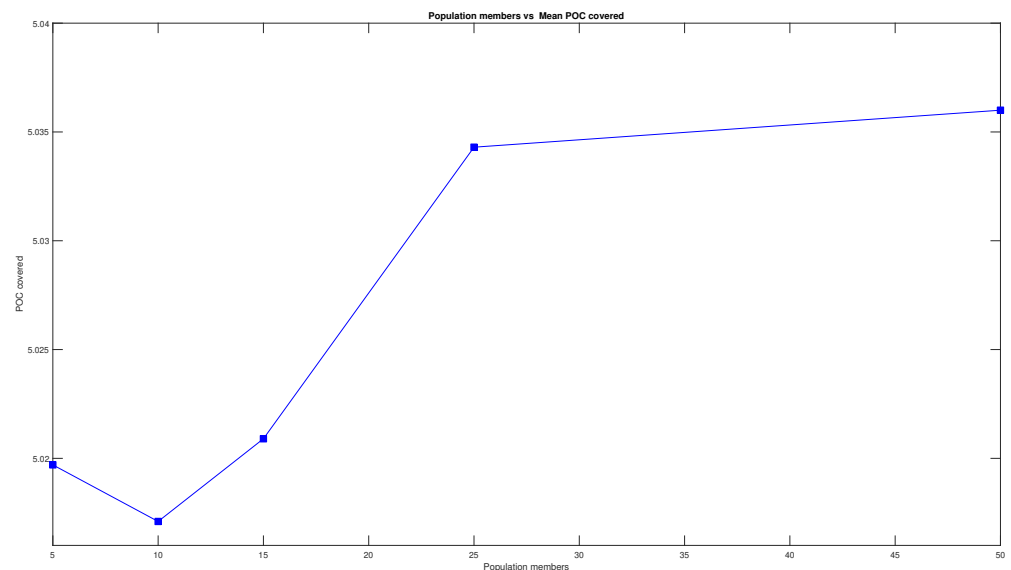


**Figure 11.** Path obtained with 50 members of the population and 5 iterations.

To better study the evolution of the POC as the number of iterations and members of the population increase, the results are plotted in graphs that can be seen in Figures 12 and 13.



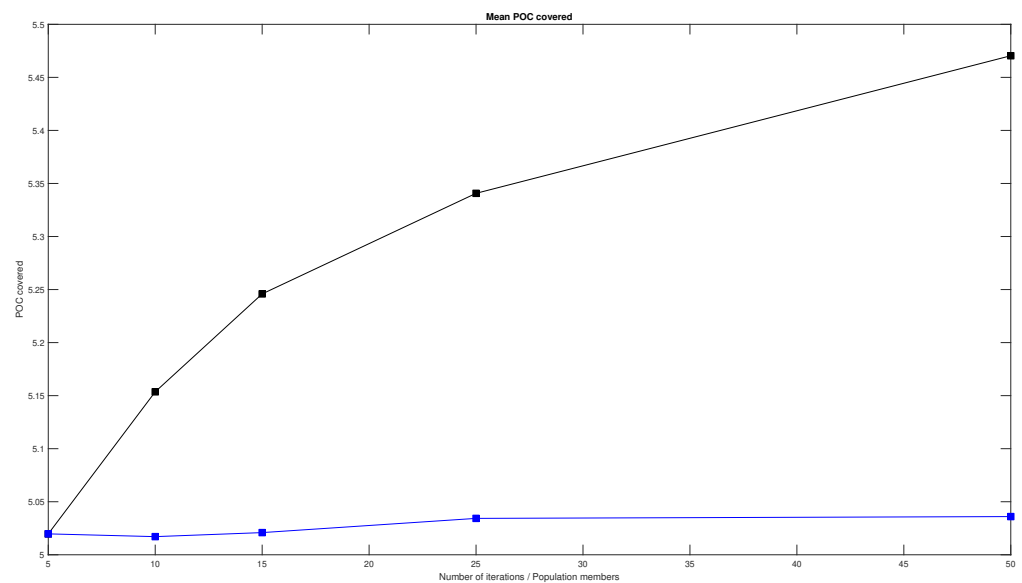
**Figure 12.** Evolution of the mean POC as the number of iterations increases.



**Figure 13.** Evolution of the mean POC as the number of members of the population increases.

The Figures show that the mean POC increases with an increment in the number of iterations and members of the population. However, the increase is not the same in both cases. To better address this difference between the two parameters, the two graphs are plotted together in Figure 14. The graph shows that the increase in the mean POC covered increases more greatly when adding more iterations than when adding more members of the population.

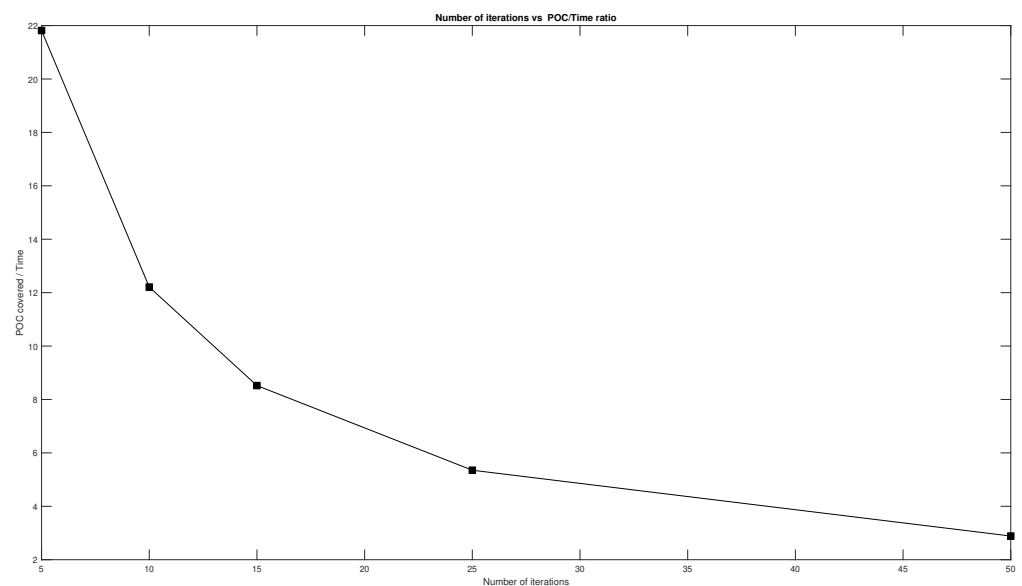
The above graphs show that, as a general rule, the results obtained from the DE algorithm improve as the number of members of the population and the number of iterations increase, but this increment becomes smaller as the DE algorithm gets closer to the optimal solution.



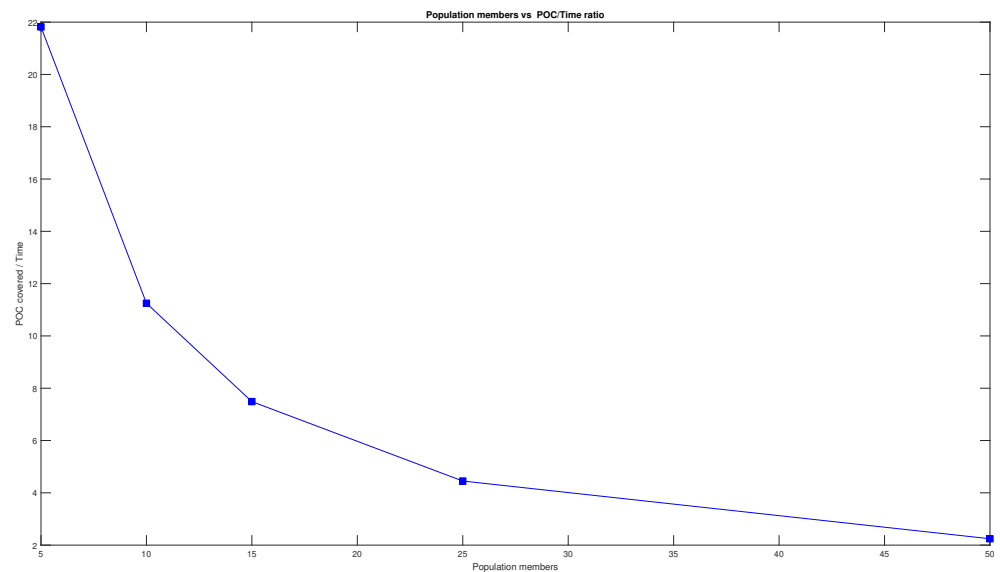
**Figure 14.** Evolution of the mean POC as the number of members of the population and number of iterations increase.

Another interesting piece of data that can be obtained from the tables above is the ratio between the POC covered and the time invested in the calculation of the coverage angle. The evolution of this ratio as the number of members of the population and the number of iterations increase can be seen in Figures 15 and 16. As can be seen in the Figures, the ratio is similar for both the increase in the number of iterations and the increase in the number of members of the population.

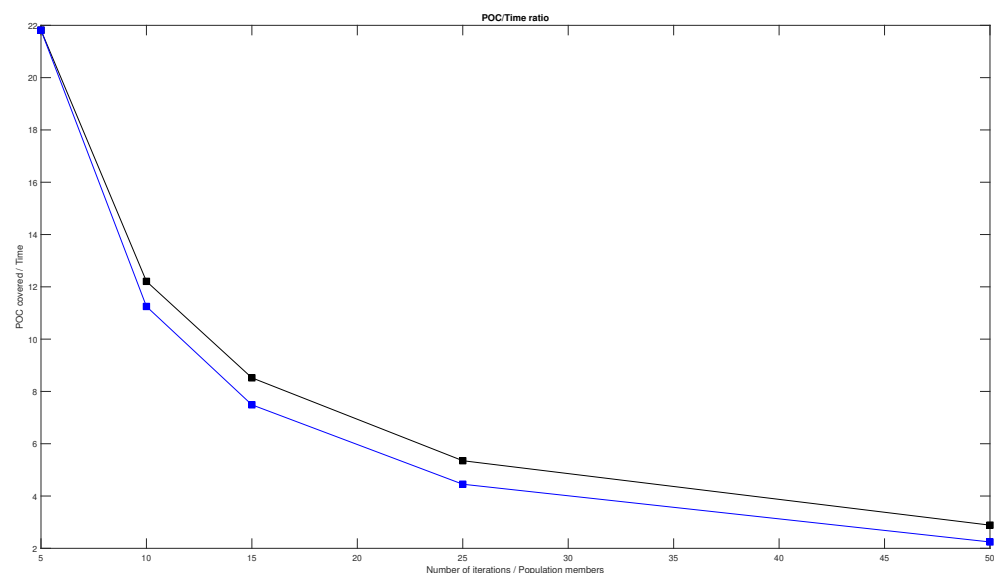
To better compare the results in both cases, the two graphs are plotted together in Figure 17. The graph shows that the ratio is in fact similar, even though the one related to the increase in the number of members of the population is slightly greater. This fact does not mean that the POC and execution times are better in this case, as has been debated before in this section, but it means that the relation between the two measures is more distinct.



**Figure 15.** Evolution of the POC/time ratio as the number of iterations increases.



**Figure 16.** Evolution of the POC/time ratio as the number of members of the population increases.



**Figure 17.** Evolution of the POC/time ratio as the number of members of the population and number of iterations increase.

These graphs show that as the number of members of the population and the number of iterations increase, the ratio between the POC covered and the time invested decreases. This means that we can choose different parameters for the DE algorithm if we want the best possible result not considering the time invested in the calculation of the best angle or a good result obtained in a short amount of time.

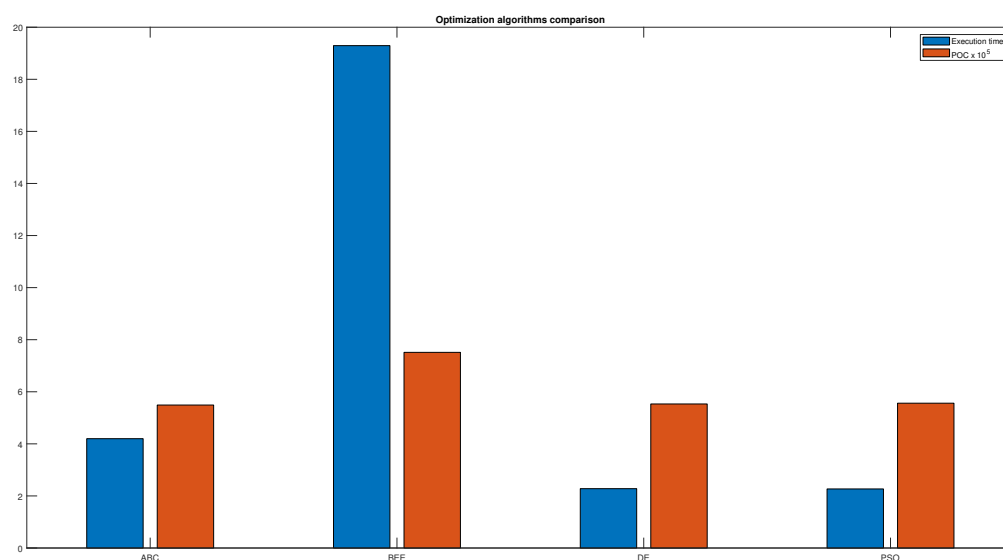
Another fact that can be inferred from the data above is that increasing the number of iterations yields a better result than increasing the number of members of the population. As it can be observed in Tables 1 and 2, the final POC covered is greater and the execution time is lower when increasing the number of iterations instead of the number of members of the population.

These results prove the great versatility of the DE algorithm in various situations.

#### *Comparison with Other Optimization Algorithms*

As it has been stated in the introduction, this operation could be performed with any optimization algorithm. To further test the results obtained in this work, the path

planning algorithm was compared with other known optimization algorithms such as particle swarm optimization (PSO) [22], the bees algorithm (BEE) [23], and the artificial bee colony algorithm (ABC) [24]. The templates for these algorithms are all taken from the work of Yarpiz [25], including the DE algorithm, to make this comparison fair. Due to that fact, the obtained results for the DE algorithm may differ with the ones shown in the tables above. The parameters for the test are 10 maximum iterations and 25 population members for all algorithms. The comparison can be seen in Figure 18:



**Figure 18.** Comparison between optimization algorithms.

The red bar indicates the POC and the blue bar represents the execution time needed to complete the optimization. As can be seen in the Figure 18, all algorithms obtain a similar POC, PSO, and DE being the least time-consuming, while the BEE algorithm obtains the highest POC with the disadvantage of needing as much as 10 times the computation time of the DE and PSO algorithms. Even though the POC obtained by the ABC algorithm is similar to the ones of the PSO and DE algorithms, the execution time is twice as much. These results prove that even though the differential evolution algorithm has been used in this work, any optimization algorithm could achieve similar results. Given that the code for the differential evolution algorithm can be obtained from Yarpiz [25], a flowchart has been included with the steps of the algorithm (Figure 8), and the ocean current data have been obtained from the public OSCAR data set [17], we believe this work is easily reproducible.

To use this algorithm in real missions, the coverage bandwidth and the flight level need to be specified to match the characteristics of the fixed-wing UAV, such as maximum altitude supported and the radius of turn, which is consequently dependent on the airspeed. These parameters need to be properly calculated and specified so that the trajectory calculated by the algorithm is feasible for the UAV.

#### 4. Conclusions

The goal of this study was to design and implement a probabilistic coverage path planning algorithm applied to search tasks in marine environments.

To generate a probability of containment map, a technique involving a marine currents data set and the use of drifters was used. After the drifters were moved by the ocean and wind currents, the probability of containment map was generated by fitting the drifters positions to a Gaussian mixture model.

The differential evolution algorithm was used to test different coverage sweep angles and obtain the best possible solution given a number of members of the population and maximum iterations. Once the best angle was obtained, a feasible coverage path was generated.



The path planning algorithm was tested with a different set of parameters to collect data about the POC covered, the time consumed, and the final coverage angles obtained. From these data, a series of plots was created to better see the evolution of the final result as the members of the population and iterations increased, and to see the evolution of the ratio between time invested in the execution of the algorithm and the results obtained.

All of the steps and the sources for the OSCAR data set and the code used in this work were detailed in this paper; thus, they should be easy to reproduce. Furthermore, the necessary parameters needed to apply the algorithm to any UAV were specified in this paper, so they can be used in real missions.

As a future work, other parameters can be considered for the cost function of the differential evolution algorithm, such as energy consumption of the UAV, and more drones can be added to create a multirobot coverage strategy.

**Author Contributions:** Conceptualization, J.M. and L.E.M.; formal analysis, F.Q.; methodology, C.A.M. and S.G.; project administration, C.A.M. and L.E.M.; software, J.M.; validation, B.L.; writing—original draft, J.M.; writing—review and editing, J.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the European Commission: Innovation and Networks Executive Agency (INEA), through the European H2020 LABYRINTH project. Grant agreement H2020-MG-2019-TwoStages-861696.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** OSCAR dataset can be obtained from [https://podaac.jpl.nasa.gov/dataset/OSCAR\\_L4\\_OC\\_third-deg](https://podaac.jpl.nasa.gov/dataset/OSCAR_L4_OC_third-deg). The Matlab function to fit the particle positions to a Gaussian mixture distribution can be downloaded from <https://www.mathworks.com/help/stats/fitgmdist.html?lang=en>. The algorithms used to perform the coverage angle optimization are all available on Yarpiz <https://yarpiz.com/> (all accessed on 5 October 2021).

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

## References

1. ATSB. The Operational Search for MH370. In *Australian Safety Transport Bureau Transport Safety Report*; Australian Transport Safety Bureau; 2017; pp. 1–440. Available online: <https://apo.org.au/node/112656> (accessed on 5 October 2021).
2. Miron, P.; Beron-Vera, F.J.; Olascoaga, M.J.; Koltai, P. Markov-chain-inspired search for MH370. *Chaos* **2019**, *29*, 041105. doi:10.1063/1.5092132.
3. Garcia-Garrido, V.J.; Mancho, A.M.; Wiggins, S.; Mendoza, C. A dynamical systems approach to the surface search for debris associated with the disappearance of flight MH370. *Nonlinear Process. Geophys.* **2015**, *22*, 701–712. doi:10.5194/npg-22-701-2015.
4. Trinanes, J.A.; Olascoaga, M.J.; Goni, G.J.; Maximenko, N.A.; Griffin, D.A.; Hafner, J. Analysis of flight MH370 potential debris trajectories using ocean observations and numerical model results. *J. Oper. Oceanogr.* **2016**, *9*, 126–138. doi:10.1080/1755876X.2016.1248149.
5. Wang, X.; Yang, L.T.; Meng, D.; Dong, M.; Ota, K.; Wang, H. Multi-UAV cooperative localization for marine targets based on weighted subspace fitting in SAGIN environment. *IEEE Internet Things J.* **2021**. doi:10.1109/JIOT.2021.3066504.
6. Xiao, X.; Dufek, J.; Woodbury, T.; Murphy, R. Uav assisted usv visual navigation for marine mass casualty incident response. In *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 24–28 September 2017; pp. 6105–6110.
7. Yilmaz, A.; Javed, O.; Shah, M. Object tracking: A survey. *ACM Comput. Surv.* **2006**, *38*, 13. doi:10.1145/1177352.1177355.
8. Ward, S.; Hensler, J.; Alsalam, B.; Gonzalez, L.F. Autonomous UAVs wildlife detection using thermal imaging, predictive navigation and computer vision. In *Proceedings of the 2016 IEEE Aerospace Conference, Big Sky, MT, USA*, 5–12 March 2016. doi:10.1109/AERO.2016.7500671.
9. Ko, A.; Lau, H.Y.K. Robot assisted emergency search and rescue system with a wireless sensor network. *Int. J. Adv. Sci. Technol.* **2009**, *3*, 69–78.
10. Chen, D.; Liu, Z.; Wang, L.; Dou, M.; Chen, J.; Li, H. Natural disaster monitoring with wireless sensor networks: A case study of data-intensive applications upon low-cost scalable systems. *Mob. Networks Appl.* **2013**, *18*, 651–663. doi:10.1007/s11036-013-0456-9.
11. Alotaibi, E.T.; Alqefari, S.S.; Koubaa, A. LSAR: Multi-UAV Collaboration for Search and Rescue Missions. *IEEE Access* **2019**, *7*, 55817–55832. doi:10.1109/ACCESS.2019.2912306.

12. Kim, M.H.; Baik, H.; Lee, S. Response threshold model based UAV search planning and task allocation. *J. Intell. Robot. Syst. Theory Appl.* **2014**, *75*, 625–640. doi:10.1007/s10846-013-9887-6.
13. Pugh, J.; Martinoli, A. Inspiring and modeling multi-robot search with particle swarm optimization. In Proceedings of the 2007 IEEE Swarm Intelligence Symposium, SIS 2007, Honolulu, HI, USA, 1–5 April 2007; pp. 332–339. doi:10.1109/SIS.2007.367956.
14. Di Franco, C.; Buttazzo, G. Energy-aware coverage path planning of UAVs. In Proceedings of the 2015 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2015, Vila Real, Portugal, 8–10 April 2015; pp. 111–117. doi:10.1109/ICARSC.2015.17.
15. Galceran, E.; Carreras, M. A survey on coverage path planning for robotics. *Robot. Auton. Syst.* **2013**, *61*, 1258–1276. doi:10.1016/j.robot.2013.09.004.
16. Ivić, S.; Crnković, B.; Arbabi, H.; Loire, S.; Clary, P.; Mezić, I. Search strategy in a complex and dynamic environment: The MH370 case. *Sci. Rep.* **2020**, *10*, 19640. doi:10.1038/s41598-020-76274-0.
17. Lagerloef, G.; Bonjean, F.; Dohan, K. *Ocean Surface Current Analysis (OSCAR) Third Degree Resolution User's Handbook*; California Institute of Technology Contract with the National Aeronautics and Space Administration and the National Oceanic and Atmospheric Administration, 2009. Available online: <https://www.cen.uni-hamburg.de/en/icdc/data/ocean/docs-ocean/users-guide-oscar-vel1992-2015.pdf> (accessed on 5 October).
18. Niiler, P. The World Ocean surface circulation. *Ocean. Circ. Clim.* **2001**, *77*, 193–204.
19. Reynolds, D. Gaussian Mixture Models. *Encycl. Biom.* **2009**, *741*, 659–663.
20. Fit Gaussian Mixture Model to Data. (n.d.). Available online: <https://es.mathworks.com/help/stats/fitgmdist.html?lang=en> (accessed on 14 May 2021).
21. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. doi:10.1023/A:1008202821328.
22. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
23. Pham, D.; Ghanbarzadeh, A.; Koc, E.; Otri, S.; Rahim, S.; Zaidi, M. *The Bees Algorithm*; Technical Note; Manufacturing Engineering Centre, Cardiff University: Cardiff, UK, 2005.
24. Karaboga, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*; Technical Report, Technical Report-tr06; Erciyes University, Engineering Faculty, Computer Engineering Department: Kayseri, Turkey, 2005.
25. Yarpiz—Academic Source Codes and Tutorials. Available online: <https://yarpiz.com/> (accessed on 30 August 2021).