A Path Planning Optimization Algorithm Based on Particle Swarm Optimization for UAVs for Bird Monitoring and Repelling – Simulation Results

Ricardo Mesquita
C-MAST - Centre for Mechanical and Aerospace Science and Technologies
Department of Electromechanical Engineering
University of Beira Interior
Covilhã, Portugal
ricardo.mesquita@ubi.pt

Pedro D. Gaspar
C-MAST - Centre for Mechanical and Aerospace Science and Technologies
Department of Electromechanical Engineering
University of Beira Interior
Covilhã, Portugal
dinis@ubi.pt

2020 International Conference on Decision Aid Sciences and Application (DASA) | 978-1-7281-9677-0/20/$31.00 ©2020 IEEE | DOI: 10.1109/DASA51403.2020.9317271

Abstract — Bird damage to orchards causes large monetary losses to farmers. The application of traditional methods such as bird cannons and tree netting became inefficient in the long run, along with its high maintenance and reduced mobility. Due to their versatility, Unmanned Aerial Vehicles (UAVs) can be very useful to solve this problem. However, due to their low autonomy it is necessary to evolve flight planning.

In this article, an optimization algorithm for path planning of UAVs based on Particle Swarm Optimization (PSO) is presented. This technique was used due to the need of an entry optimization algorithm to start the initial tests. The PSO algorithm is a simple and has few control parameters while maintaining a good performance. This path planning optimization algorithm aims to manage the drone's distance and flight time, applying optimization and randomness techniques, to be able to overcome the disadvantages of other systems. The performance of the proposed algorithm was tested in a tree case simulation that represents all the possible cases.

Keywords — Unmanned Aerial Vehicles, Path Planning, Optimization Algorithm, Meta-heuristic, Bird Damage

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs), commonly known as drones, are electronic systems with low maintenance and high versatility. This technology has been developed over the years and nowadays there are autonomous drones based on path planning and waypointing with commercial standards applied in different areas such as military search and rescue and agriculture [1], [2].

Despite continuous evolution of the techniques and technologies to control bird damage in agriculture, the loss of fruit and trees remains a long-term and costly problem. Birds flock, such as of Starling and Magpie, as shown in Figure 1, cause high losses over the years to farmers by destroying the trees and damaging fruits, making them susceptible to diseases, which can lead to a decrease in production and quality [3]. The most used repelling methods are the loudspeaker with sounds of danger emitted by the same species or predatory animals, bird cannons, tree netting and planned planting and harvesting. All techniques mentioned have the same problems that are high maintenance and in the long run become inefficient, because birds get used to it. In order to solve these, we resort to bird monitoring already applied in airports or science studies and UAVs that can be helpful due to its mobility, reaching multiple affected areas without predictable patterns.

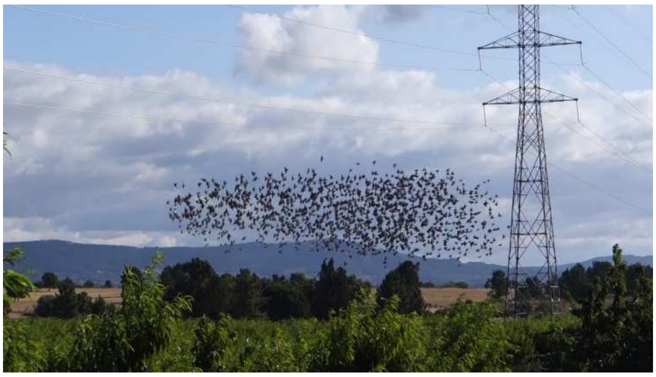

Figure. 1. Flock of black starling on a peach field.

When building a drone to perform a specific job is necessary to consider weather, typology, control, lift weight and battery capacity since all these factors will influence the flight duration. Supposing that all these components are well dimensioned and from the hardware the drone cannot have more flight time, it is necessary to start optimizing the flight process, i.e., the path planning to ensure greater effectiveness during flights. Meta-heuristic optimization algorithms are simple techniques inspired in physical phenomena, animal behaviors or evolutionary concepts that have become very popular over the last decades to solve problems in different fields [4]. Due to their simplicity, these algorithms have been applied when a fast solution is required. Particle Swarm Optimization (PSO), the most well-known meta-heuristic, swarm based, bio-inspire optimization algorithm. It is very simple to use and implement due to the small number of control parameters, presenting itself as a good entry level optimization method with fit performance results to test the path planning algorithm in all scenarios. This technique has already been applied in various fields of industry such as network weights and network structure for artificial neural network, reactive power and voltage control, and ingredient mix [5].

In this article, we present and test a path planning optimization algorithm to be used with a bird monitoring and repelling system. This device consists in two separate parts: (1) a device coupled to a drone that will generate audio and light in a random way and (2) a network of sensors that will detect the movement of birds and send the data to the main computer. The data will be analyzed by the optimization algorithm and a flight path will be generated according to the bird movement. The focus of this work is the path planning optimization algorithm and not the complete solution.

II. Related Work

Most of the work developed in path planning is related to the Vehicle Routing Problem (VRP) or with is variant Vehicle Routing Problem with Drones (VRPD). This problem aims to determine a set of optimal routes performed by vehicles with limited capacity to serve a given set of customers. Many models, algorithms and heuristics have been developed for it [6]. The VRPD is a recent problem that has been receiving a lot of work due to the interest of companies like DHL, Amazon, and Alibaba. This is a perfect example of a drone flight duration problem and the need to improve path planning according to the task required. Tseng et al. [7] show the concept of battery-operated and flight with recharging. Firstly, an empirical study of energy consumption to determine battery performance was performed, considering various flight scenarios and the commercial model 3DR. Next, the problem of flight planning with recharging optimization for drones was studied, where the goal was to complete a tour mission for a set of sites of interest. In the end the solution is implemented and tested in different case studies. Recently, more work related with agriculture and drones has been developed. Rabello et al. [8] present a mobile app to optimize the drone flight in a precision agriculture scenario. The Android platform utilizes Google Tools and for the optimization an algorithm based on recursive auctions. This app was developed to generate a drone path planning based on waypoints with a predefine distance in the chosen area.

Farmers need to spray daily different farm blocks. Using a single drone becomes an impossible task due to endurance and battery change. So, Li et al. [9] use PSO algorithm to optimize flight path in UAVs groups. This work focus on optimize the flight paths of the whole UAVs group with minimum make-span instead of minimizing the total flight distance.

As described, some work has been developed in path planning and flight optimization. However, the novelty of this study is related to the application of the proposed algorithm to bird damage problem. Additionally, to authors' knowledge it is the first algorithm introducing the concept of random waypointing to avoid patterns, together with optimization techniques to ensure that the drone travels the maximum or minimum distance. In this work we decided to optimize the path planning with a Swarm technique, especially the PSO, due to its simplicity and efficiency for the first tests of the global algorithm.

III. Particle Swarm Optimization

PSO was first proposed by Eberhart et al. [10] and it is a method for optimization of continuous nonlinear functions. This evolutionary computation technique was developed to simulate a simplified social system and has been used for approaches that can be used across a wide range of applications or specific requirements [5].

The optimization algorithm initialized with a population (*swarm*) defined as *N* of random solutions called particles, that have the dimension of the problem define as *dim*, and to each solution is assigned a randomized velocity. Each *particle* keeps track of its best solution in the problem space (fitness) and the corresponding coordinates, this value is called *pbest*. The overall global fitness of the swarm is also tracked, and it is called *gbest*.

Every iteration defined as *it* changes the velocity defined as *v* of each *particle* toward its *pbest* and *gbest* locations. Velocity is weighted by two different random numbers in the interval [0,1] defined as *r1* and *r2* and two constants named *c1* and *c2*. The random numbers control the acceleration, and the constants control the stochastic acceleration terms towards *pbest* and *gbest.*

In the proposed algorithm, different from the original equation, the velocity is also controlled by an inertia weight defined as *w* that provides a balance between global and local exploration and exploitation, and results in fewer iterations on average to find sufficiently optimal solution. In the original work, authors created a maximum velocity named *Vmax* that serves as a constraint to control the global exploration ability of the swarm.

At the end, it is necessary to verify if all the particles are inside the *bound* dimension of problem. Figure 2 shows the pseudo code of PSO algorithm that was used in the proposed algorithm.

```
Initialize the swarm with N and dim
Initialize c1, c2, w, itmax
while it < itmax:
        for each particle
            Calculate the fitness of each particle
            Update the pbest and gbest
        end for
        for each particle
            r1 = rand
            r2 = rand
            v  =  w*v+c1*r1*(pbest-particle)+  c2*r2*(gbest-
particle)
            particle = particle + v
            if particle < lower bound or particle > upper bound
                      Initialize a new particle
            end if
        end for
        it = it +1
end while
```

Figure. 2. Pseudo code of PSO algorithm.

IV. Algorithm Architecture

In order to develop the proposed optimization algorithm, it was necessary to study the problem and the tools that were needed to solve it. For the bird damage in agriculture problem, it is necessary constant application of the repelling systems, so the algorithm needs to minimize the flight time between sensors and maximize it in the most affected areas. But birds can detect pattern and learn how to avoid them, so is also required to the algorithm the ability to read the data from the sensors and create different numbers of random waypoints according to the areas.

Since the beginning we decided to use the Mission Planner Home, created by Michael Oborne as our ground control station due to its open source, point-and-click waypoint system and configuration. To use this software the algorithm needs to generate a file of WAYPOINTS (.waypoints) that consists of a geographic coordinated system (latitude and longitude) and height above takeoff. Figure 3 represents the main steps of the optimization algorithm to generate the waypoint file that will be used to path plan drone flights.

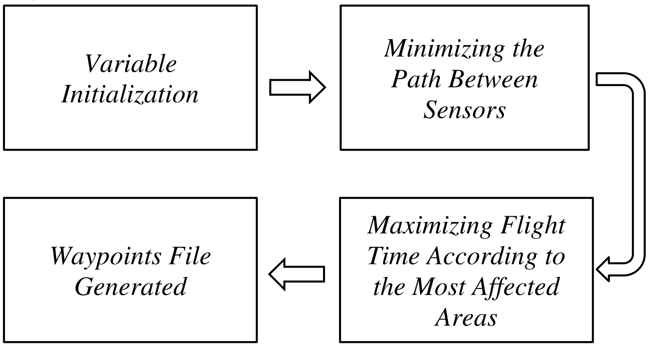

Figure. 3. Path planning optimization algorithm main steps.

A. *Variable initialization*

This algorithm has the objective to be applied to any bird or field, so it is necessary some specifications and variables, such as the geographic coordinates of take-off, landing and sensors. The coordinates need to be described in latitude and longitude, respectively and the file with the detection data needs to be in the same order as the algorithm. This system does not provide feedback with the drone, so it is necessary to enter the maximum flight distance. It is also required the maximum and minimum distance from which the waypoint will be generated according to the sensor.

B. *Haversine Formula*

Several formulas have been tested to measure the distance between two geographic coordinates, including the Google Maps Platform Distance Matrix API. We ended up choosing Haversine Formula due to its simplicity and precision.

The Haversine Formula is an important equation in navigation, giving great-circle distances between two points on a sphere from their longitudes and latitudes [11]. To apply this method is necessary the geographic coordinates of each point, represented as *lat1, lat2* and *lon1, lon2* respectively and the radius of the earth defined as *r*. Equation *(1)* and *(2)* represents the Haversine Formula, the distance between the two points is defined as *d*.

$$c = \sqrt{\sin^2\left(\frac{\text{lat1-lat2}}{2}\right) + \cos(\text{lat1}) * \cos(\text{lat2}) * \sin^2\left(\frac{\text{lon1-lon2}}{2}\right)} \qquad (1)$$

$$d = 2*r*\arcsin(c) \qquad (2)$$

C. *Path between sensors*

Depending on the type of field and the position of the sensors, the drone must flight according to the needs. For that, it is necessary that the algorithm receives the data and establish the shortest path to save battery for the areas next to the sensors where birds are.

To calculate the fastest route, the PSO to minimization is used where each particle contains a value corresponding to a geographic coordinate of a sensor and the objective function is the sum of the distances, using Haversine Formula, between sensors, the origin and final position. In the end, this function will send the minimum distance and the order of the sensors that the drone needs to fly by. In the event of a sensor does not have recent detections, the algorithm will eliminate it in that flight. Figure 4 represents two cases of flight between sensors, one with optimization represented as *a)* and the other without optimization represented as *b)*. The path of the first case scenario has a total distance of 216.3 m, while the path of the second case scenario has 268.3 m, which represents a reduction of 52 m.

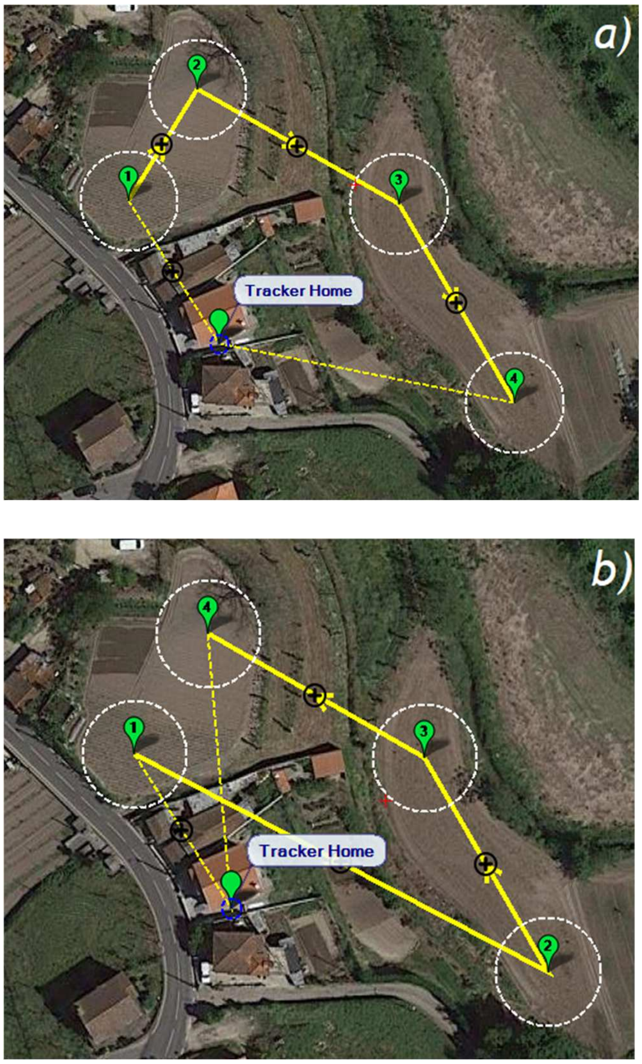

Figure. 4. a) With optimization function b) Without optimization function.

D. Waypoint distribution in affected areas

After planning the order of the sensors, the proposed algorithm will remove the distance between the sensors from the total distance and will determine a maximum number of points according to the minimum distance between them, which is established through a cycle that executes the difference between the actual final distance and the one introduced by the user. Thus, the maximum number of points possible for the drone will be calculated. Then, random waypoints around each sensor will be created, generated based on the minimum and maximum distances to the sensor and their number varies according to the number of detections. Figure 5 represents the pseudo code of the function that generates random geographic coordinates.

```
lb = [value non-zero]
ub = [this number changes the distance that the waypoint will be created]
for up to number of waypoints of the sensor
    while distance to sensor > maximum or distance to sensor < minimum
        for until 2 (latitude, longitude)
            value = lb+(ub-lb)*rand()
            waypoint = first four values of the geographic coordinates of the sensor + value
        end for
        distance to sensor = Haversine Function of waypoint and sensor
    end while
end for
```

Figure. 5. Pseudo code random waypoints function.

PSO will be applied again but in this case for maximization. Each particle corresponds to the value of a new random waypoint and for the objective function the Haversine Formula is used again where the distance from the first and last points are tested with the position of the sensor itself. In the end, the sum of all distance is performed and tested with the maximum flight distance. In case that the two values are the same (with an error), a file will be generated with the flight positions and the geographical coordinates of each waypoint. Otherwise, the algorithm generates new points and runs the maximization PSO again.

V. CASE STUDIES

To test the proposed algorithm, a flight mission was simulated in a peach field in the region of Orjais, Covilhã, in Portugal. Table 1, shows the latitude and longitude of each sensor in the simulation.

TABLE I. COORDINATES OF EACH SIMULATION SENSOR.

Sensor	Latitude	Longitude
Sensor 1	40.34037440	-7.37956170
Sensor 2	40.34070150	-7.38012490
Sensor 3	40.34114710	-7.38064530
Sensor 4	40.34075460	-7.38121390
Sensor 5	40.34030080	-7.38072570
Sensor 6	40.33995320	-7.38012490

Three case studies were then chosen to test the proposed algorithm in all conditions, where initially the same number of waypoints were assigned to each sensor (*case a)*), then a random number (*case b)*) and in the end, zero detections were assigned to some sensors (*case c)*). Table 2 shows the detection ratio that was attributed to each sensor, per case study. This is the ratio between the number of birds detected individually and the sum of all sensors, however due to these tests being a simulation the values are random.

TABLE II. DETECTION RATIO IN EACH SENSOR PER CASE.

Case	a)	b)	c)
Sensor1	0.1667	0.1000	0.0000
Sensor2	0.1667	0.050	0.2500
Sensor3	0.1667	0.2500	0.0000
Sensor4	0.1667	0.3000	0.4000
Sensor5	0.1667	0.1000	0.2000
Sensor6	0.1667	0.2000	0.1500

For all the cases, a total flight distance of 1500 meters was used, with 25 meters of acceptance error by the algorithm. The maximum and minimum values for the generation of random waypoints were 15 and 2 meters respectively, a lower bound of 0.0000001 and upper bound of 0.001 to latitude and 0.0001 to longitude. In both PSO, constants *c1* and *c2* were 2, the inertia weight of 0.4, the initial velocity values were generated randomly. Only 5 particles were used for each algorithm so that they could process faster. The maximum number of iterations was different due to the complexity of each problem, establishing 50 and 200 iterations for the maximization and minimization optimization algorithm, respectively.

VI. RESULTS

The proposed path planning optimization algorithm were developed in Python coding language and executed in the PyCharm IDE developed by JetBrains. Tests were performed on a Windows 10 of 64-bit computer, i7-6700HQ CPU and 16GB of RAM. Due to the heuristic nature of the algorithms, they were tested individually sixty times each.

For each simulation, the flight distance data in meters, the total number of waypoints, the execution time in seconds and the number of iterations that the code had to re-run to find a solution within the acceptance range were collected and the average was calculated, represented in table 3. As shown in the previously mentioned table, all case study presents a similar result to the reference distance. The rest of the test parameters are identical in cases *a)* and *b)*, with *c)* having a much higher value in all. This happens especially for the time and iteration parameters, due to the global architecture of the algorithm, and the random nature of the PSO, where it is necessary to execute more steps when there are areas without birds causing a greater distribution number of waypoints per sensor.

TABLE III. DETECTION RATIO IN EACH SENSOR PER CASE.

Case	Distance [m]	Total Waypoints	Iterations	Time [sec]
a)	1503,52	82,483	14,967	39,521
b)	1504,465	83,883	13,767	42,036
c)	1505,761	100,983	28,383	184,07

We must emphasize that the overall processing time of the path planning optimization algorithm, despite all its complexity, is quite good for a practical case. In all cases having a maximum time of 390.2175 seconds processing case c) and a minimum of 3 seconds in case a).

VII. CONCLUSION

Since agriculture presents itself as a crucial sector, it is necessary to understand and eliminate all related problems. One of these problems arises from bird damage since they can destroy various types of crops ranging from fruits to grains [12]. The common techniques due to their low mobility and randomness become ineffective in the long run. UAVs have high versatility and low maintenance, which makes them potential solutions to these problems. Its biggest problem is its low energy capacity, so it is necessary to improve the efficiency of the type of flight by planning the trajectory.

Heuristic techniques can optimize complete systems while maintaining the simplicity and efficiency of other algorithms. One of these techniques is Particle Swarm Optimization and is used in the proposed algorithm to find the best path through the distance.

The basic architecture of the algorithm presented in this paper can be divided in four steps. Initiation of parameters, where the geographic coordinates of takeoff, landing and sensors are introduced. Then, the path between sensors is minimized, ensuring that the drone does not waste time between sensors. Subsequently, random points are generated around the sensors and their path is maximized. At the end, a file is created to be read by the drone's controller.

This is a recent work that needs to be improved but it has already shown satisfying results in the three simulations, having in all cases presented a good average in distance and execution time. Future work will be related to the optimization of the algorithm, more specific in the test of other optimization techniques and the improvement of the function that generates random waypoints. These approaches should improve the execution time and quality of the algorithm. Field tests must also be carried out where the capacity and autonomy of the batteries must be tested, along with the efficiency to solve the bird damage problem.

ACKNOWLEDGMENTS

This research work is funded by the PrunusBot project - Autonomous controlled spraying aerial robotic system and fruit production forecast, Operation No. PDR2020-101-031358 (leader), Consortium No. 340, Initiative No. 140, promoted by PDR2020 and co-financed by the EAFRD and the European Union under the Portugal 2020 program.

The authors thank the opportunity and financial support to carry on this project to Fundação para a Ciência e Tecnologia (FCT) and R&D Unit "Centre for Mechanical and Aerospace Science and Technologies" (C-MAST), under project UIDB/00151/2020.

REFERENCES

[1] Y. Karaca *et al.*, "The potential use of unmanned aircraft systems (drones) in mountain search and rescue operations," *Am. J. Emerg. Med.*, vol. 36, no. 4, pp. 583–588, 2018, doi: 10.1016/j.ajem.2017.09.025.

[2] U. R. Mogili and B. B. V. L. Deepak, "Review on Application of Drone Systems in Precision Agriculture," *Procedia Comput. Sci.*, vol. 133, pp. 502–509, 2018, doi: 10.1016/j.procs.2018.07.063.

[3] J. L. Elser *et al.*, "Measuring bird damage to three fruit crops: A comparison of grower and field estimates," *Crop Prot.*, vol. 123, no. May, pp. 1–4, 2019, doi: 10.1016/j.cropro.2019.05.010.

[4] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[5] R. C. Eberhart and Y. Shi, "Particle swarm optimization: Developments, applications and resources," *Proc. IEEE Conf. Evol. Comput. ICEC*, vol. 1, pp. 81–86, 2001, doi: 10.1109/cec.2001.934374.

[6] Z. Wang and J. B. Sheu, "Vehicle routing problem with drones," *Transp. Res. Part B Methodol.*, vol. 122, pp. 350–364, 2019, doi: 10.1016/j.trb.2019.03.005.

[7] C.-M. Tseng, C.-K. Chau, K. Elbassioni, and M. Khonji, "Autonomous Recharging and Flight Mission Planning for Battery-operated Autonomous Drones," no. March, 2017, [Online]. Available: http://arxiv.org/abs/1703.10049.

[8] T. E. Rabello. Andre, Brito. Robison, Favarim. Fabio, Weitzenfeld. Alfredo, "Mobile System for Optimized Planning to Drone Flight applied to Precision Agriculture," 2020.

[9] X. Li, Y. Zhao, J. Zhang, and Y. Dong, "A hybrid PSO algorithm based flight path optimization for multiple agricultural UAVs," *Proc. - 2016 IEEE 28th Int. Conf. Tools with Artif. Intell. ICTAI 2016*, pp. 691–697, 2017, doi: 10.1109/ICTAI.2016.0107.

[10] R. E. James Kennedy, "Particle Swarm Optimization," 1995.

[11] C. N. Alam, K. Manaf, A. R. Atmadja, and D. K. Aurum, "Implementation of haversine formula for counting event visitor in the radius based on Android application," *Proceedings of 2016 4th International Conference on Cyber and IT Service Management, CITSM 2016*. 2016, doi: 10.1109/CITSM.2016.7577575.

[12] M. B. Hannay *et al.*, "Bird species and abundances in fruit crops and implications for bird management," *Crop Prot.*, vol. 120, no. September 2018, pp. 43–49, 2019, doi: 10.1016/j.cropro.2019.02.015.