Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

School of Engineering and Sciences

Bioinspired Search and location of Air Pollutant Sources based on a Bayesian Heuristic + Dragonfly Algorithm.

A thesis presented by

Andrés Fernando García Calle

Submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Master of Science

in

Engineering Sciences

Monterrey, Nuevo León, December, 2019

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

School of Engineering and Sciences

The committee members, hereby, certify that have read the thesis presented by Andrés Fernando García Calle and that it is fully adequate in scope and quality as a partial requirement for the degree of Master of Science in Engineering Sciences.

Dr. Luis Eduardo Garza Castañon
ITESM
Principal Advisor

Dra. Adriana Vargas Martínez
ITESM
Committee Member

Dr. David Alejandro Sotelo Molina
ITESM
Committee Member

Dr. Rubén Morales Menéndez
Dean of Graduate Studies
School of Engineering and Sciences

Monterrey Nuevo León, December, 2019

Declaration of Authorship

I, Andrés Fernando García Calle, declare that this thesis titled, ”Bioinspired Search and location of Air Pollutant Sources based on a Bayesian Heuristic + Dragonfly Algorithm.” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this dissertation is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Andrés Fernando García Calle
Monterrey, Nuevo León, December, 2019

©2019 by Andrés Fernando García Calle
All Rights Reserved

Dedication

The present work is dedicated to Wilson García, Aida Calle and Carlos García for always being my main support.

Acknowledgements

The culmination of this work has been an important step in my life. Many people have helped me achieve my goals:

To my brother and parents for their advice and support.

To Dr. Luis Eduardo Garza who knew how to guide me to carry out the project.

To my master's director, Dr. Oliver Probst, for giving me the opportunity to study in the Master's Degree in Engineering Sciences and to be aware of the situations that we students go through.

To the future Dr. Kenneth Palacios who with his classes motivated me to continue in the field of research.

To Dr. Ismael Minchala who supported me in my career thesis and advised me to continue with my master's studies.

To Dr. Adriana Vargas and Dr. David Sotelo, for the corrections and recommendations that helped me improve the project.

To Edisson, Yael, Eduardo, Llivi, Heriberto and all the members of #YaelElEscarabajo who gave me their friendship.

To Memo, Cristina, Carlos Katt, Víctor, Salvador, César, Fermín, Armando, Rosario, Eduardo, Chui, Jorge, members of the Robotics laboratory. I am very grateful for their advice and friendship.

Finally, I would like to thank ITESM and CONACYT who financed my master's degree.

Bioinspired Search and location of Air Pollutant Sources based on a Bayesian Heuristic + Dragonfly Algorithm.

by

Andrés Fernando García Calle

Abstract

In nature there are many types of interaction between individuals. These behaviors have been optimized in millions of years and their main objective is the survival of the species. The objective of interacting with other individuals is to solve difficult problems, problems that a single individual would solve inefficiently. The problem aborted by this thesis is the search for and location of air pollutant sources. A SO_2 plume is simulated and placed in a region to perform the search. To collect contaminant samples, the dynamics of 2 unmanned aerial vehicles are simulated. The algorithm runs under a real-time simulation environment. A 2-dimensional Hammersley sequence is used in an exploration stage. Clustering by K-means and a Greedy Search are also used to solve the Travelling Salesman Problem at the Hammersley points. In this first stage an attempt is made to find a measure of the concentration of air pollutants that exceeds a certain threshold. In an exploitation stage, the Unmanned Aerial Vehicles (UAVs) will track the source approaching this point. A probability map is used to modify the behavior of the Dragonfly Optimization Algorithm. The probabilistic map acts as the heuristic of the system to locate the source of air pollution. To check the efficiency of the proposed strategy it is compared with 3 other algorithms: Greedy Search, Greedy Search with cooperation and the Dragonfly Algorithm without modifications. All these approaches are tested in the same simulated environment, without obstacles for UAVs. Efficiency is measured with respect to the maximum level of concentration of contaminants found, the distance to the source of contamination, the time taken to find the maximum concentration of contaminants and the UAVs path until that time. Finally, a one-factor ANOVA is performed to check that the responses are statistically better.

Symbols

This document is incomplete. The external file associated with the glossary ‘symbols’ (which should be called `output.sls`) hasn’t been created.

Check the contents of the file `output.slo`. If it’s empty, that means you haven’t indexed any of your entries in this glossary (using commands like `\gls` or `\glsadd`) so this list can’t be generated. If the file isn’t empty, the document build process hasn’t been completed.

You may need to rerun LaTeX. If you already have, it may be that TeX’s shell escape doesn’t allow you to run makeindex. Check the transcript file `output.log`. If the shell escape is disabled, try one of the following:

- Run the external (Lua) application:

  ```
  makeglossaries-lite.lua "output"
  ```

- Run the external (Perl) application:

  ```
  makeglossaries "output"
  ```

Then rerun LaTeX on this document.

This message will be removed once the problem has been fixed.

List of Figures

1.1 Cities with air pollution problems due PM10, PM2.2 and O3 in México [11]. 1

2.1 Signed package format in MAVLINK V2 protocol. 7
2.2 Greedy Search Flowchart 10
2.3 Dragonfly behavior in nature [22] 11
2.4 Pollution dispersion from a stack due the wind, turbulent mixing and gravity [13]. 14
2.5 Search area divided in $n \times m$ sub cells [23] 17

3.1 Project Methodology 19
3.2 The simulated UAV is the Scorpion 280-RATE model 20
3.3 Plume Shape after 20 minutes of release 21
3.4 Map boundaries established at the time of take-off 23
3.5 How to reference the cells in the probabilistic map 23
3.6 Flowchart of the basic Navigation System 25
3.7 Map updated by a Greedy Search algorithm 27
3.8 Flowchart to calculate a new position on the probabilistic map 28
3.9 Cooperative update of probability map. 29
3.10 The enclosed area is visited many times by UAVs. 30
3.11 The step vector is the sum of all components (Eq. 2.6). 30
3.12 Scheme of Methods used by the algorithms 31
3.13 35 points from a 2D Hammersley sequence. 32
3.14 Solution of 2 Travel Salesman Problems. Each number is the Hammersley position in each sequence. 33
3.15 Updated map using Hammersley sequences, clustering and TSP. 34
3.16 Flowchart for Dragonfly Algorithm 35
3.17 Dragonfly Flow Chart that shows the Heuristic component 36

4.1 UAV take-off Zones. Each pair of dots of the same color symbolizes UAVs location for take-off. 37
4.2 Plume location on the map 38
4.3 Plume simulated in an area of $167x300\ meters^2$ 38
4.4 Percentages of detection of high levels of concentration of pollutants in the air 40
4.5 Proximity vs. time of the Greedy Search Algorithm 41
4.6 Proximity vs. time of the Cooperative Greedy Search Algorithm 41
4.7 Proximity vs. time of the Dragonfly Algorithm 42

4.8 Proximity vs. time of the Heuristic Dragonfly Algorithm 42
4.9 Boxplot for time dispersion of results 43
4.10 Boxplot for proximity dispersion of results 44
4.11 Boxplot for air pollutant concentration of results 45
4.12 Boxplot for path dispersion of results 46

List of Tables

1.1 Summary of the application of UAVs for air quality monitoring 3

2.1 Format for a MAVLink 2 packet [33] 8

3.1 Parameters of the simulated multicopter 20

4.1 Statistical values for time responses 43
4.2 Statistical values for proximity responses 44
4.3 Statistical values for concentration dispersion of responses 45
4.4 Statistical values for path responses 46
4.5 Single-factor ANOVA results 46
4.6 Characteristics of the proposal against similar works 47

Contents

Abstract v

Symbols vi

List of Figures viii

List of Tables ix

1 Introduction 1

1.1 Problem statement and Motivation 2
1.2 Similar Works 2
1.3 Hypothesis 5
1.4 General objective 5
1.5 Specific objectives 6
1.6 Research Questions 6
1.7 Contribution 6
1.8 Composition 6

2 Theoretical Framework 7

2.1 MAVLINK protocol 7
  2.1.1 MAVLINK packaging 7
2.2 UAV interactions 9
  2.2.1 UAVs Interactions 9
2.3 Greedy Search 10
2.4 Dragonfly Optimization 11
2.5 Hammersley sequence 13
  2.5.1 Van der Corput sequences 13
  2.5.2 Halton sequence 13
  2.5.3 Hammersley Sequence 13
2.6 K-Means Clustering 13
2.7 Pollutant Plume Model 14
  2.7.1 Advection-Diffusion PDE 14
  2.7.2 Boundary Conditions 15
  2.7.3 The source 15
  2.7.4 Diffusivity in Low Pressure 16
  2.7.5 Analytical solution 16

2.8 Probabilistic Map 16
  2.8.1 Search Area 17
  2.8.2 Probability map based on one detection or no detection events 17

3 Methodology 19
3.1 Virtual UAVs 19
3.2 Probabilistic Map and Plume simulation 20
  3.2.1 Plume Simulation 20
  3.2.2 Plume Mapping 21
  3.2.3 Probabilistic Map 22
  3.2.4 Map delimiters 22
  3.2.5 Cell subdivisions 23
  3.2.6 Bayesian-Based Map 24
3.3 Navigation System 25
  3.3.1 Greedy Search as a Navigation Algorithm 26
  3.3.2 Cooperative Greedy Search 28
  3.3.3 Dragonfly Navigation Algorithm 30
  3.3.4 Exploration stage 31
  3.3.5 Exploitation Stage 34
3.4 Heuristic Dragonfly 35

4 Simulation and Results 37
4.1 Parameters and Assumptions 39
4.2 Detections and no Detections 39
4.3 Air Pollutant Source Proximity 40
4.4 Distribution of Results 42
4.5 Analysis of results 46

5 Conclusions 49
5.1 Future work 50

Bibliography 53

Chapter 1

Introduction

Nowadays, it is not strange to hear about environmental pollution. Daily, the news talk about the pollution of forests, sea, rivers, air, etc. Pollution harms agriculture, livestock, people and animals. For instance, air pollution causes diseases such as cerebral stroke, asthma, cancer, and heart or lung conditions. In many cities, entities such as the World Health Organization (WHO) conduct environmental monitoring to verify levels of air pollutants. In Mexico, cities like Toluca, Monterrey, Mexico city among others, show high levels of air pollutants concentrations. According to the statistics, every year there are around 9300 deaths occasioned by air pollution. The 3.2% Gross National Product in Mexico is spent on environmental issues. Figure 1.1 shows the air pollutants present in Mexico, according the Greenspace organization.

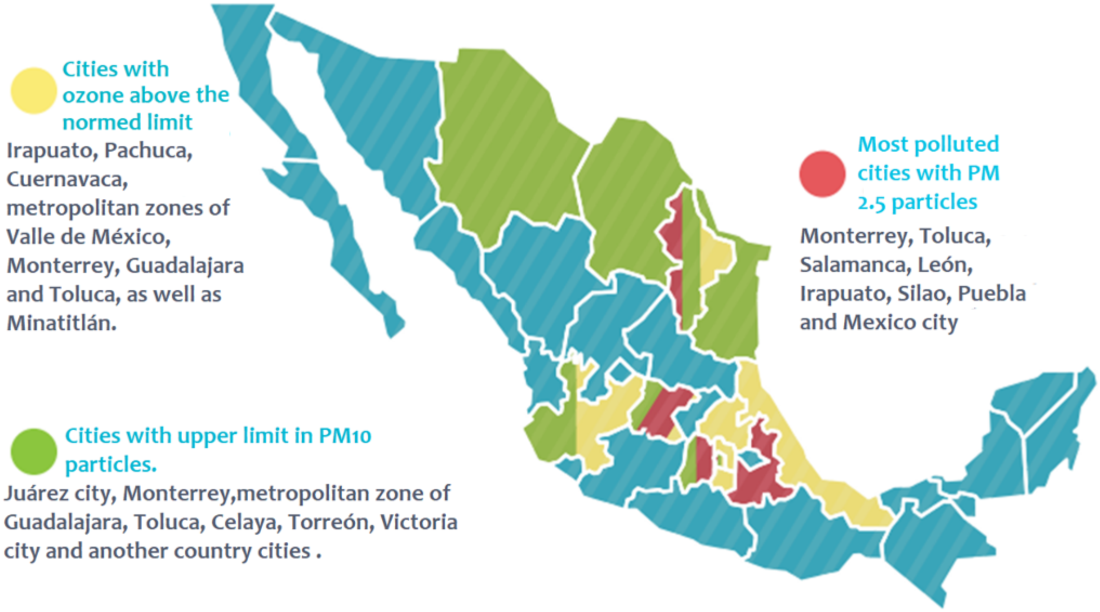

Cities with ozone above the normed limit
Irapuato, Pachuca, Cuernavaca, metropolitan zones of Valle de México, Monterrey, Guadalajara and Toluca, as well as Minatitlán.
Cities with upper limit in PM10 particles.
Juárez city, Monterrey,metropolitan zone of Guadalajara, Toluca, Celaya, Torreón, Victoria city and another country cities .
Most polluted cities with PM 2.5 particles
Monterrey, Toluca, Salamanca, León, Irapuato, Silao, Puebla and Mexico city

Figure 1.1: Cities with air pollution problems due PM10, PM2.2 and O3 in México [11].

The monitoring systems are used to alert for high pollutant concentrations in the air. In Monterrey, environmental monitoring is carried out by Sistema Integral de Monitoreo Ambiental (SIMA), with 13 towers that are distributed in strategic points of the city.

The stations have sensors that provide very accurate measurements of the concentration of pollutants, temperature, magnitude and wind direction, etc. Although the key monitoring task is performed, the stations are constrained to achieve fixed spatial sampling. To expand the applications of static monitoring platforms is necessary to take information from several points with a non-static system. Balloons and UAVs are options for doing this task. Platforms based on UAVs can complement static monitoring systems. By having greater versatility, can get more information about the dispersion of pollutants in the air. In addition to above, they have some advantages such as: being able to mobilize in the 3 dimensions of space, low cost, access to difficult or dangerous places for people. A problem connected with the monitoring of air pollutants is related to the search and location of sources of pollution. This a challenging problem, due mainly to the highly dynamic environment, the uncertainty and the wide areas to search. UAVs instrumented with adequate sensors and guided by intelligent algorithms can be used as an effective tool to deal with the problem of search and locate pollutant sources.

1.1 Problem statement and Motivation

The key factor to succeed in the location of pollutant sources is the efficient use of flight time of UAVs, in order to take more samples and cover more areas. In this work, is described the design and testing of cooperative intelligent heuristics which can be used to guide the navigation of 2 UAVs, in order to search and locate the source of air pollutants.

1.2 Similar Works

This section presents a table showing various works in which UAVs have been used for similar purposes to the project. Different approaches are taken into account, such as interactions between UAVs, types of monitoring that have been carried out in a real or simulated environment, target tracking techniques, etc. One of the main differences between the proposal and the related work is that the tests were simulated in real-time. The 2 main stages of the algorithm are exploration and exploitation. In these stages a search area is sought until a high level of concentration of pollutants is found and then until the highest concentration point is approached. Other characteristics are that the search area is large and quadcopters are used to make the sensing of contamination levels. Below are the main objectives of each project, as well as a brief description of the test platform and any striking features of the project.

Table 1.1: Summary of the application of UAVs for air quality monitoring

Author (et.al), year	**Objective**	**UAV Characteristics**	**Notes**
V. Smídl 2012 [29]	Tracking an atmospheric release of pollution with two autonomous navigate UAVs.	Two UAVs and simulations	The platform pretends to complement ground station networks for monitoring the air quality. There is a comparison of wind direction between real measurements and ALADIN numerical model.
J. Han 2012 [12]	Detect nuclear radiation with a platform based on multiple UAVs.	Simulations and fixed-wing UAV with wingspans about 48 in.	The contour mapping of the nuclear radiation is simulated. 2 UAVs formations were discussed: in a normal nuclear reactor operation and when there is a radiation leakage
R. Kristiansen 2012 [18]	Discuss UAVs formations for monitoring industrial emissions.	Simulated	Discuss UAVs formations for monitoring industrial emissions.The UAV formation configuration and the UAVs operation strategies are analyzed. Some concepts for guidance and control of UAVs are given too.
S. Zhu, 2013 [31]	Seek a unknown scalar source field with multiple UAVs.	Simulated	The multiple UAVs cooperation is based on a leader-follower strategy and a gradient descent algorithm.
A. Khan, 2014 [16]	Propose strategies to measure the impact of cooperation between multiple UAVs.	Simulated	The individual map is updated with the sensing information, then this information is shared with the rest of UAVs and then group actions are taken.
S. Li 2014 [20]	Study robotic tracking of dynamic plume front modeled by advection-diffusion equation	Simulated	A single robot and a group of cooperating robots is used to tracking the dynamic pollution plume. The transport model of the pollution source and the gradient measurements are considered in tracking control design.
B. Di, 2015 [6]	Regulated the network topology and optimize the surveillance horizon in a multiple UAV platform.	Simulated	A distributed receding horizon optimization algorithm regulate the horizon boundaries.

Table 1.1 continued from previous page

M. Alvarado 2015 [1]	Develop a methodology for characterizing blasting pollution plumes in near-real time.	Small fixed-wing and multi-rotor copter, dust sensor SHARP GP2Y10	PM2.5 and PM10 particle concentration were sampled. The collected data is streamed during flight. A correlation of the dust sensor utilized and a industry grade dust sensor is shown.
J.J. R, 2016 [25]	Propose a new control architecture for multi-UAV missions.	Simulated in: Multiple Mini-Robot Simulator (MMRS)	The proposal architecture has three layers: the action layer,the task layer and the mission layer.
F. Koohifar, 2017 [17]	Locate a moving radio frequency transmitter with a UAV swarm.	Simulated	The system uses path planning algorithm based on a finite receding horizon.
M. Rossi, 2016 [26]	Develop a new board for environmental monitoring that can be embedded in any mobile carrier.	DJI hexacopter with 80-cm diameter.	The work evaluate the pollutant sensors sensitivity in presence of propellers turbulent air. Two experiments were developed: a flight with a isotropic alcohol source and a flight over a kitchen.
J.J. Cassano, 2016 [4]	Scan the details of the air-sea ice-ocean coupling using 5 fixed wing UAV over the Terra Nova Bay polynya, Antarctica.	Wingspan of 3.6m, maximum speed of 22/33 m/s, RF communication of 900 MHz and a Iridium satellite phone modem	The data from the coordinated UAVs flights give a 3 dimensional data set of the atmospheric state (air temperature, humidity, pressure, and wind) and surface skin temperature over Terra Nova Bay.
Y. Qu, 2017 [24]	A cooperative positioning approach for multiple UAVs flight.	Simulated	This is a fault tolerant cooperative positioning system.The approach uses the azimuth angle and a angle relative to a reference.
V. Sharma, 2017 [27]	Secure the network connectivity of a UAV swarm through an efficient strategy to select a controller and autonomous relaying.	Simulated	The work presents a system that ensure the connectivity between the swarm members. The approach is based on a neural model.
P. Li, 2017 [15]	Track pollutants in the environment by using a swarm of UAVs.	Simulated in QRSim.	The UAV swarm reconfigure itself when pollution is detected.

Table 1.1 continued from previous page

C. Greatwood, 2017 [9]	Atmospheric sampling for characterize methane concentrations and their isotopic composition from Ascension Island, in the middle of Atlantic.	Octocopter with X-8 configuration, typical battery capacity of 533 Wh, temperature and humidity sensors, autopilot with Pixhawk and ArduCopter v3.1.5	A Sampling platform based on a Small Unmanned Air System (SUAS) is used for sample the atmosphere, also the temperature and relative humidity were measured. The measurements were transmitted to a ground station.
T. Zhu, 2018 [32]	Search targets located in probabilistic positions.	Simulated	Use a discretization of searching areas by grid division. A spiral fly model is used for path optimization.
S. Benders 2018 [2]	Improve the path tracking performance through a proposed technique based in wind adaptive path planning.	Simulation and Skywalker X8 fixed-wing model equipped with position, attitude, velocity over ground and airspeed sensors.	The paths are updated with estimates of the local wind velocity. The platform uses in-flight wind velocity estimator. 2 scenarios were simulated for the wind: a scenario with constant air density and other with turbulent wind.
N. Yungaicela 2018 [30]	This work propose an algorithm for air pollutant source localization.	A quadrotor Q-ball (Quanser) equiped with a QuaRC-powered Gumstix embedded computer.	A probabilistic method was used to compute a heuristic source location. This technique, together with a gradient based search algorithm, was developed to localize the pollutant source. The algorithm was tested in a simulated polluted environment.

1.3 Hypothesis

An algorithm based on the flying dragon behavior can be an efficient cooperative strategy to guide 2 UAVs in the search and location of pollutant sources.

1.4 General objective

Design and testing of a biomimetic strategy for search and location of pollutant sources with 2 cooperative UAVs.

1.5 Specific objectives

- Design and implementation of the biomimetic strategy for search and location of pollutant sources with 2 UAVs.
- Design of simulated scenarios to test the efficiency of search and location algorithms.

1.6 Research Questions

Throughout the work it is intended to answer the following questions:

- Is it possible to combine the information taken by each quadcopter to improve the platform's search algorithm based on one UAV?
- Is it possible that 2 UAVs can find the source of an air pollutant in wide areas?
- Does the use of cooperative flight strategies significantly improve the location time of a polluting air source?

1.7 Contribution

Main contributions of this research work are:

1. Design of a new cooperative strategy to guide the navigation of UAVs, based on a dragon fly algorithm and a heuristic map
2. Demostrate the effectivity of the proposed algorithm against similar approaches.

1.8 Composition

The content of the following chapters is summarized below:

- Chapter 2 contains the theoretical framework of the project. Here it will be explained the algorithms, techniques, and models that were used to develop the project.
- Chapter 3 explains the project methodology. It shows every step taken to develop the proposed algorithm, using the concepts mentioned in chapter 2.
- Chapter 4 presents the experiments, the assumptions, and parameters used in the scenarios and the results.
- Chapter 5 presents the discussion, conclusions, recommendations and the future.

Chapter 2

Theoretical Framework

This chapter shows the fundamentals concepts related to navigation, optimization, UAV construction and others.

2.1 MAVLINK protocol

MAVLink is a binary telemetry protocol for communications on aerial platforms with UAV. This protocol is used through XML files. This feature allows you to use the protocol with several programming languages such as C, java, python, etc. Telemetry data streams are sent in a multicast design. Configuration aspects are carried out through more secure protocols with retransmission, such as the Mission Protocol or the Parameter Protocol. There are 2 versions of the protocol: MAVLINK 1 and 2. All the commands of the first version are compatible with the second.

2.1.1 MAVLINK packaging

The format of a MAVLINK 2 package is illustrated in the figure 2.1. Every package component is explained in the table 2.1.

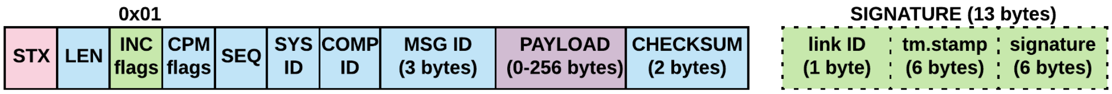


Figure 2.1: Signed package format in MAVLINK V2 protocol.

Table 2.1: Format for a MAVLink 2 packet [33]

Byte Index	Data	Content	Explanation
0	STX	Packet start marker	Beginning of a new package
1	LEN	Payload length	Indicates length of the following payload section
2	INC flags	Incompatibility flag	For MAVLink compatibility
3	CMP flags	Compatibility flag	For MAVLink compatibility
4	SEQ	Packet sequence number	Used to detect packet loss
5	SYS ID	System ID	ID of system (vehicle)
6	COMP ID	Component ID	Used to differentiate components in a system
7 to 9	MSG ID	Message ID	Used to decode data back into message object
n to n+10	PAYLOAD	Payload	Message data
n+11 to n+13	CHECKSUM	Checksum	X.25 CRC for message
n+12 to n+26	SIGNATURE	Signature (optional)	Signature to ensure the link is tamper-proof.

The bits that make up the payload are those in which the UAV navigation orders will be placed. Below is an example of a Mavlink message in the Python language.

```python
def goto_position_target_global_int(self, Location):
    """
    Send SET_POSITION_TARGET_GLOBAL_INT command to request the vehicle fly
     to a specified LocationGlobal.
    """
    msg = self.FlyController.message_factory.
   set_position_target_global_int_encode(
        0,
        0, 0,
        mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT_INT,  # frame
        0b0000111111111000,  # type_mask (only speeds enabled)
        Location.lat * 1e7,  # lat_int - X Position in WGS84 frame in 1e7
  * meters
        Location.lon * 1e7,  # lon_int - Y Position in WGS84 frame in 1e7
  * meters
        Location.alt,  # alt - Altitude in meters in AMSL altitude
        0,
        0,
        0,
        0, 0, 0,
        0, 0)
    # send command to vehicle
    self.FlyController.send_mavlink(msg)
```

Mavlink_code_for_move_an_UAV_to_a_desired_Position.py

2.2 UAV interactions

2.2.1 UAVs Interactions

A team is defined as a collection of loose, controlled and distributed objects in space. In the case of UAVs , a team consists of several drones and sometimes a human operator. The interactions of a team are induced by common goals or tasks. There are various degrees of interaction in a team: coordinated, cooperative and collaborative [28].

Cooperative Interactions

In this team, all members have specific target functions, but also try to optimize the overall cost function. Private or global functions are weighted with a value of $\mathbf{0} \leq \mathbf{w} \leq \mathbf{1}$, where the following cases of w can be presented for the private objective function:

- $\mathbf{w} = \mathbf{1}$ means that there's no team action.
- $\mathbf{0} < \mathbf{w} < \mathbf{1}$ means that there is a degree of cooperation with team members.
- $\mathbf{w} = \mathbf{0}$ implies that every individual target function is in strict coordination with the overall target function.

Coordinate Interactions

In this case, the UAVs of a team share the objectives to be met. This means that all members will try to optimize the same cost function. If there are several cost functions these can be weighted and put together to form a global objective function. There are no conflicts of interest between team members. Some team members may not have objective functions, they are considered usable resources to meet the overall objective. All team members are obligated to participate in some task that helps to fulfill the common goal. The team is distributed in the space but operates as a single unit.

Collaborative Interactions

Collaborative teams can result from task allocation or resource allocation. Each team member attempts to optimize their cost function, taking into consideration the overall cost function and avoiding conflicting tasks. This type of interaction requires a negotiation or arbitration protocol. Private target functions are necessary tasks for the global target function. The more coupled the tasks of the team members, the more difficult it is to carry out the common task. Cooperative or coordinated operations are necessary, as there may be task conflicts. If a task cannot be solved by a member, the member may choose not to participate and join an alternative team that is more in line with the overall objective.

2.3 Greedy Search

This section explains concepts about Greedy Search for swarm robotic navigation. The goal of navigation is to find a target in a limited area. The approach of this navigation is directed towards the use of UAVs. The use of this kind of vehicle avoids putting people at risk, in addition to providing more efficient results.

One of the most common search strategies is Greedy Search. This algorithm is based on following a gradient until some condition is satisfied. The algorithm always chooses the path that brings the UAV closest to its target. This approach has the advantage that operations with high computational costs are not necessary. If swarms of UAVs are used, communication between everyone isn't strictly necessary. Each UAV can locate the target with its collected information, using the Gradient Descent technique [8]. This increases the algorithm robustness if any UAVs are lost, also reduces the probability of stagnation at local minimums or maximums.

One of the important aspects of the search is the collection of close information. To move the UAV to a new position it must first map all its alternatives from its current position. Once all the information has been obtained, they are compared and the measure with the best aptitude is selected. It can be said that the UAV is transformed into a mobile sensor. Then, the UAV goes to the position that has the best aptitude and repeats the cycle, until a completion criterion is met. A completion criterion can be that all UAVs arrive at the same position. In figure 2.2 a flowchart about the greedy search is showed.

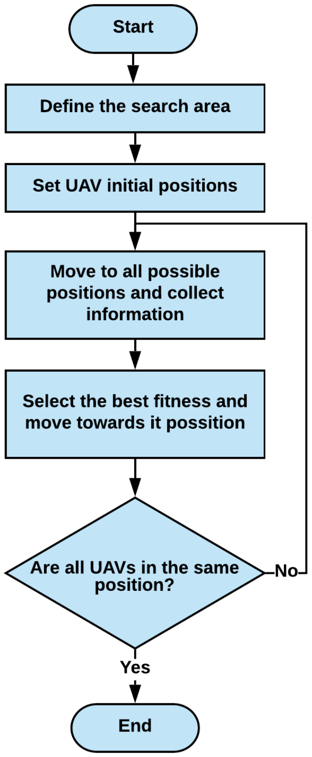

Start
Define the search area
Set UAV initial positions
Move to all possible positions and collect information
Select the best fitness and move towards it possition
Are all UAVs in the same position?
No
Yes
End

Figure 2.2: Greedy Search Flowchart

2.4 Dragonfly Optimization

In nature it's common to see groups of individuals to solve problems. These problems are often difficult for a single individual but working as a team efficiently resolves them. For species survival, interactions have been perfected over millions of years. These include colonies of ants that cooperate to get food, schools of fish that alert each other of predators, packs of wolves that have an efficient hunting system, among others [22].

The observation of this type of behavior gives rise to several bio inspired algorithms. Swarm Intelligence is a field dedicated to studying these behaviors. In this field, algorithms have been developed to mimic the interactions of biological groups and thus solve several problems. The Dragonfly Optimization Algorithm is based on these insects, but it is also influenced by PSO (Particle Swarm Optimization) and ACO (Ant Colony Optimization). The survival of dragonflies depends on factors such as the search for food and the evasion of predators. The behaviors of swarms of dragonflies are shown in the figure 2.3. The swarm behavior of the aforementioned algorithm allows to exceed local minimums or maximums.

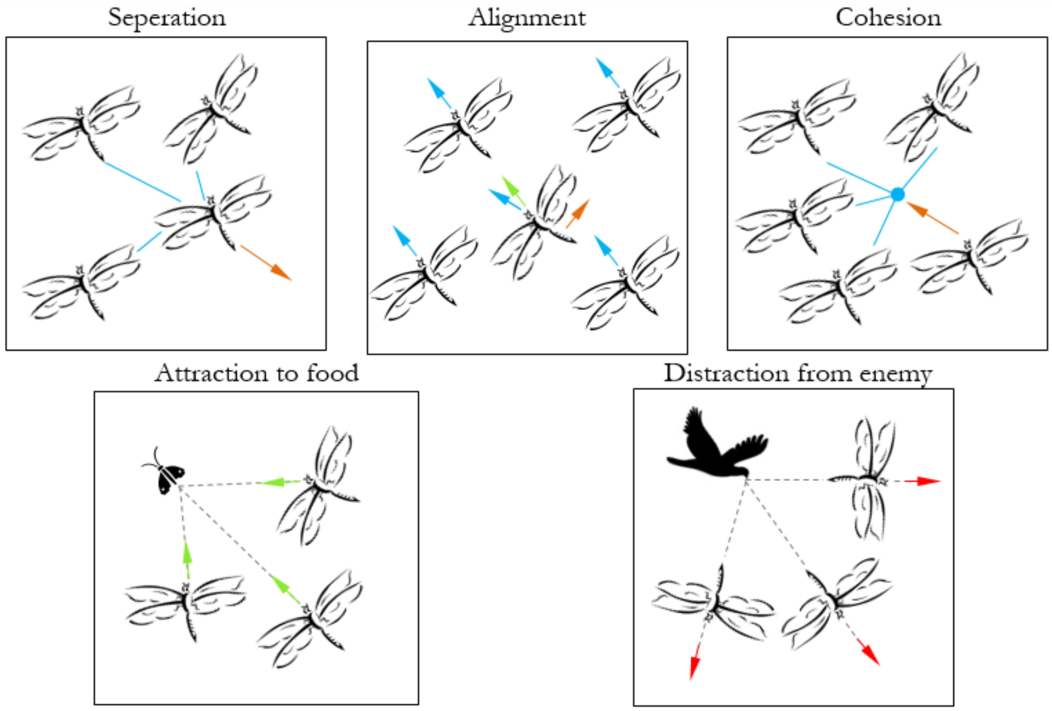


Figure 2.3: Dragonfly behavior in nature [22]

The previous figure shows each form of interaction of the dragonflies in a swarm. These behaviors, and their mathematical modeling, are described below:

1. **Separation:** This behavior serves to prevent members of the swarm from colliding. There is a tendency for the individual to separate from the entire swarm.

$$S_i = -\sum_{j=1}^{N} X - X_j \tag{2.1}$$

2. **Alignment:** It refers to the speed of adaptation of individuals with respect to others in the neighborhood.

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N} \tag{2.2}$$

3. **Cohesion:** It is the tendency of each individual to go towards the group centroid.

$$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X \tag{2.3}$$

4. **Attraction to food:** It allows the individual to approach the place where there is food.

$$F_i = X^{+} - X \tag{2.4}$$

5. **Stay away from predators:** Allows you to get away from certain points.

$$E_i = X^{-} + X \tag{2.5}$$

Where:
i represents the i-th swarm individual.
N is the number of individuals in the swarm.
X is the position of the current individual.
X_j is the position of the j-th neighbor of the swarm.
V_j is the speed of the j-th neighbor of the swarm.
X^{+} is the position of the food.
X^{-} is the position of the enemy.
The step vector is calculated with:

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \tag{2.6}$$

s, a, c, f, e and w are the weights that determine the behavior of the swarm. Finally, the position vector is calculated as:

$$X_{t+1} = X_t + \Delta X_{t+1} \tag{2.7}$$

To perform the exploration phase with randomness and stochastic behavior, it is necessary to travel the search space with a random path [22]. In the section 2.5 we describe the process to meet this objective using Hammersley sequences. In addition, for this stage it is considered to use step vectors with a Lévy distribution in case the Hammersley sequence does not approximate the algorithm to the solution. In the case of using the Lévy distribution, the position of the dragonflies is:

$$X_{t+1} = X_t + \text{Lévy}(d) \cdot X_t \tag{2.8}$$

Where:

$$\text{Lévy}(d) = c \cdot \frac{r_1 \sigma}{|r_2|^{\frac{1}{\beta}}} \tag{2.9}$$

Being r_1 y r_2 random numbers with uniform distribution in $[0, 1]$, c a scalar and σ the result of the following expression:

$$\sigma = \left(\frac{\Gamma(1 + \beta \cdot sin(\pi\beta/2))}{\Gamma(\frac{1+\beta}{2})\beta \cdot 2^{\frac{\beta-1}{2}}} \right)^{1/\beta}$$

A high cohesion coefficient and a low separation coefficient are required for the exploitation phase.

2.5 Hammersley sequence

To define a Hammersley sequence it is necessary to explain about van der Corput sequences and Halton sequences first [5].

2.5.1 Van der Corput sequences

A van der Corput sequence is the simplest one-dimensional low-discrepancy sequence. This sequence is over the interval from 0 to 1. The van der Corput sequence uses the representation of positive integers on a given basis (b-ary representation). This sequence is defined as:
For $n \geq 1$, let:

$$n - 1 = \sum_{j=0}^{s} a_j b^j \tag{2.10}$$

be the dyadic expansion of $n - 1$. Then we set the sequence $(x_b(n))$ as:

$$x_b(n) = \sum_{j=0}^{s} a_j(n) b^{-j-1} \tag{2.11}$$

Where a_j is the $j - th$ digit in the $b - ary$ representation of the integer n.

2.5.2 Halton sequence

Halton sequences is an expansion of van der Corput sequences, but for higher dimensions. For this series we take arbitrary $b_1, ..., b_s$ integer numbers that are coprime with each other. This numbers will be the bases to form the van der Corput sequences resulting in:

$$x_n = (x_{b1}(n), ..., x_{bs}(n)) \tag{2.12}$$

2.5.3 Hammersley Sequence

A Hammersley sequence is nothing more than a Halton sequence, in which replacing a dimension by a linear sweep. The sequence results as:

$$x_n = \left(x_{b1}(n), ..., x_{bs-1}(n), \frac{n}{N} \right) \tag{2.13}$$

Being N the length of the sequence [19].

2.6 K-Means Clustering

When a big amount of time has passed, the UAVs next positions tend to be the same, hence the Hammersley sequence is used to avoid the converge of objective points. The resulting points are distributed on the map, but not assigned to a specific UAV. As the initial positions of the UAV can be different, it is necessary to divide the k Hammersley points into 2 groups. The K-means clustering is a non-deterministic and unsupervised algorithm that can perform the previous task [7]. Every cluster obtained from k-means algorithm will maximize the similarity between their members and minimize the similarity with the other clusters. The basic K-means algorithm is given as [21]:

1. Choose 2 points as the initial centroids.
2. Determine distance between each Hammersley point and the cluster centroid.
3. Reassign data point to new cluster with minimum distance.
4. Get the centroids of the new clusters.
5. Repeat till convergence criteria is meet.

The stop algorithm criteria is given when the clusters have shown no change or the sum squared error (Eq. 2.14) reach a small value.

$$SSE = \sum_{i=1}^{C} \sum_{j=1}^{C_i} (||X_i - Y_j||)^2 \tag{2.14}$$

Where $||X_i - Y_j||$ is the Euclidean distance between X_i and Y_j. C_i is the total members in the $i-th$ clusters. C is the number of cluster centroids.

2.7 Pollutant Plume Model

This section describes the model of an air pollutant plume. The modeled plumes have a point source, of the type shown in figure 2.4. This model begins with the resolution of the Partial Differential Equation (PDE), which represent the pollution plume [13].

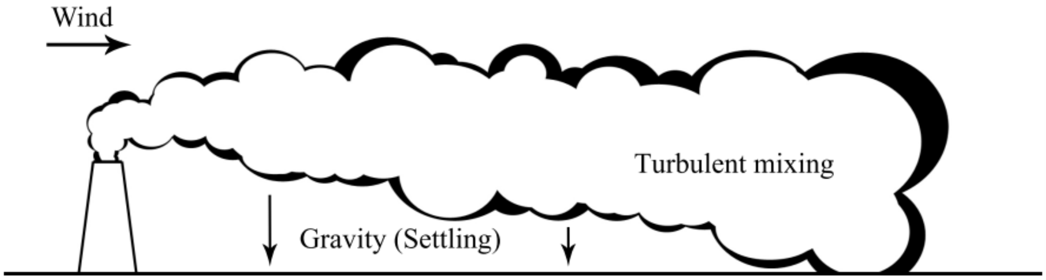


Figure 2.4: Pollution dispersion from a stack due the wind, turbulent mixing and gravity [13].

2.7.1 Advection-Diffusion PDE

In order to analyze the dispersion of pollutants in the environment, dispersion and advection models will be necessary. The differential equation that models these phenomena is governed by the following principles:

- Pollutant Concentration
- The relationship between flow and density

Defining $c(\vec{X},t)[kg/m^3]$ as the concentration of the mass in a given time $t[sec]$, in addition to $\vec{X}=(x,y,z)\in\mathbb{R}[m]$ as the position, we have that the equation of concentration is:

$$\frac{\partial c}{\partial t}+\nabla\cdot\vec{f}=s \tag{2.15}$$

Where $\vec{f}(\vec{X},t)[kg/m^2 s]$ is the flow of contaminant mass due to diffusion and advection and $s(\vec{X},t)[kg/m^3 s]$ is the source of contamination. $\vec{f}$ is the term that has the components to model the advection and diffusion of the pollution plume. For the diffusion of one substance into another, Fick's law is used:

$$\vec{f_D}=-\vec{D}\nabla c \tag{2.16}$$

Where $\vec{D}=(D_x,D_y,D_z)$ has the diffusion coefficients in the respective axis. These coefficients are usually in the function of the position. The advection component is given by the dragging of the fluid by the wind:

$$\vec{f_A}=-c\vec{u} \tag{2.17}$$

Where $\vec{u}=(u_x,u_y,u_z)[m/s]$ is the wind velocity. Adding up the two components of f, and replacing them in equation 2.15 we have:

$$\frac{\partial c}{\partial t}+\nabla\cdot(c\vec{u})=\nabla\cdot(\vec{D}\nabla c)+s \tag{2.18}$$

2.7.2 Boundary Conditions

Couple of constraints must be taken into account in solving the 2.18 equation:

- If the z-axis is considered as the ground, the dispersion only happens in $z\geq 0$.
- The concentration tends to 0 when is analyzed in a far field.

Then:

$$u_z c-D_z\frac{\partial c}{\partial z}=0 \ \ at \ \ z=0 \tag{2.19}$$

If we consider $u_z=-u_{set}$ as settling velocity for the particulates and W_{dep} as a deposition coefficient that captures the effect of total flux of contaminants penetrating the ground, we have the Robbin Boundary Condition:

$$-u_{set}c-D_z\frac{\partial c}{\partial z}=-W_{dep}c \tag{2.20}$$

2.7.3 The source

A Dirac delta function is used to model the source of pollution, allowing it to be modeled as a point source in space:

$$s(\vec{X},t)=Q\cdot\delta(x-x_s)\delta(y-y_s)\delta(z-z_s) \tag{2.21}$$

Where: $\delta(\cdot)$ is the Dirac delta function, $\vec{x}_s=(x_s,y_s,z_s)$ is the source position and $Q[kg/s]$ is the total output of the source in unit time.

2.7.4 Diffusivity in Low Pressure

There are several methods for calculating the coefficient of the diffusion of one substance in another. In this work the Fuller method will be used:

$$D_{AB} = \frac{0.00143T^{1.35}}{PM_{AB}^{0.5}\left[(\sum v_A)^{1/3} + (\sum v_B)^{1/3}\right]^2} \tag{2.22}$$

Where:
T is the temperature ($298^o K$).
P is the pressure (1.02 bar).
M_{AB} is the harmonic average of molecular weights [g/mol].
$\sum v$ is the Molecular volumes of diffusion.

2.7.5 Analytical solution

To obtain the analytical solution of the differential equation we need some assumptions:

- The wind is in the direction of the x-axis: $\vec{u} = (u, 0, 0)$.
- The source position is: $\vec{X} = (0, 0, H)$
- The contaminant cannot penetrate the ground.
- Concentrations tend to zero in the far-field.
- The solution for negative values of x is not physically relevant.
- The diffusion coefficients are: $D_x = D_y = D_z = D$
- The advection effect is much greater than the diffusion effect: $D_x = 0$

Using all the assumptions we can reduce equation 2.18 in:

$$u\frac{\partial c}{\partial x} = D\frac{\partial^2 c}{\partial^2 y} + c\frac{\partial^2 c}{\partial^2 z} + Q\delta(x)\delta(y)\delta(z - H) \tag{2.23}$$

That has as solution:

$$c(r, y, z) = \frac{Q}{4\pi ur} exp\left(\frac{-y}{4r}\right)\left[exp\left(-\frac{(z-H)^2}{4r}\right) + exp\left(-\frac{(z+H)^2}{4r}\right)\right] \tag{2.24}$$

This solution is associated with a Gauss distribution with variance $\sigma^2 = 2r$.

2.8 Probabilistic Map

The work develop by [23] shows the obtention of a probability map for the location of chemical plumes.

2.8.1 Search Area

To facilitate calculations, the search area is rectangular. This area is divided into $n \times m$ sub cells with length L_y and width L_x. A vector of cells $C = [C_1, \cdots, C_M]$ covering the area of interest is also required, where $M = m \cdot n$. Let $a \in [1, m]$ count over cells in x direction. Let $b \in [1, n]$ count over cells in y direction. For mapping cells with respect to a and b we need:

$$i = a + (b-1)m \tag{2.25}$$

For the inverse mapping, with respect to i, we need:

$$a(i) = rem(i-1, m) + 1 \tag{2.26}$$

$$b(i) = int(i-1, m) + 1 \tag{2.27}$$

Where $rem(n, m)$ is the remainder of n/m, and $int(n, m)$ is the greatest integer less than or equal to n/m. The figure 2.5 show cellular subdivision of the target area.

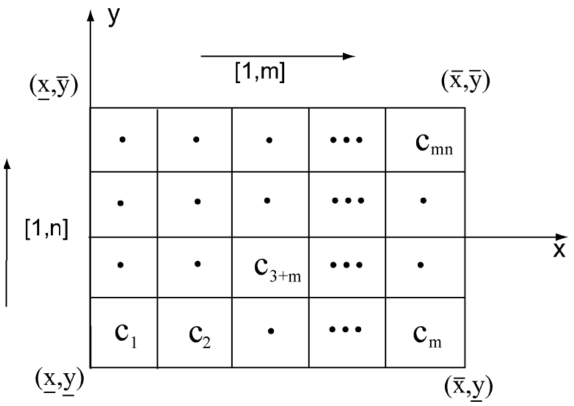

y
[1,m]
$(\underline{x},\bar{y})$
$(\bar{x},\bar{y})$
[1,n]
x
c_{mn}
c_{3+m}
c_1
c_2
c_m
$(\underline{x},\underline{y})$
$(\bar{x},\underline{y})$

Figure 2.5: Search area divided in $n \times m$ sub cells [23]

2.8.2 Probability map based on one detection or no detection events

To obtain the probability map it is necessary to define some probabilities and events:

- $S_{ij}(t_l, t_k)$ is the probability that there is a source in C_i that released one chemical filament at time t_l, given that the chemical is in C_j at time $t_k (t_k > t_l)$.

  $$S_{ij}(t_l, t_k) = \frac{e^{-\frac{(x_j - x_i - v_x(t_l,t_k))^2}{2(t_k - t_l)\sigma_x^2}} e^{-\frac{(y_j - y_i - v_y(t_l,t_k))^2}{2(t_k - t_l)\sigma_y^2}}}{2\pi(t_k - t_l)\sigma_x \sigma_y} L_x L_Y \tag{2.28}$$

  Where v is the velocity vector.

- $\beta_{ij}(t_0, t_k)$ is the probability that there is a source in cell C_i, given that there is detectable chemical in cell C_j at time t_k.

  $$\beta_{ij}(t_0, t_k) = \frac{1}{k} \sum_{l=0}^{k-1} S_{ij}(t_l, t_k) \tag{2.29}$$

- $\gamma_{ij}(t_0, t_k)$ is the probability of not detecting the chemical in cell C_j at time t_k due to the continuous release from a source in cell C_i.

  $$\gamma_{ij}(t_0, t_k) = \prod_{l=0}^{k-1}(1 - \mu S_{ij}(t_l, t_k)) \tag{2.30}$$

  Where μ denote the probability of detecting chemical given that there is detectable chemical in the cell.

- A_i is the event that there is a source in cell C_i.
- $D(t_k)$ is the event that there is a chemical detection at time t_k in cell C_j.
- $\bar{D}(t_k)$ is the event that there is no detection at time t_k in C_j.
- $B(t_{k-1})$ is a sequence of detection and no detection events from time t_0 to t_{k-1}.
- $\alpha_i(t_{k-1})$ is the source probability map based on the sequence of detection and no detection events between times t_0 and t_{k-1}.

To update $\alpha_i(t_k)$ we assume that $B(t_{k-1})$ and $D_j(t_k)$ are independent events.
If there's a detection in C_j:

$$\alpha_i(t_k) = P(A_i|B(t_{k-1}), D_j(t_k)) = M\alpha_i(t_{k-1})\beta_{ij}(t_0, t_k) \tag{2.31}$$

If there's not a detection in C_j:

$$\alpha_i(t_k) = P(A_i|B(t_{k-1}), \bar{D}_j(t_k)) = \frac{M\alpha_i(t_{k-1})\gamma_{ij}(t_0, t_k)}{\sum_M^{i=1} \gamma_{ij}(t_0, t_k)} \tag{2.32}$$

Chapter 3

Methodology

This chapter presents a description of the strategy used to meet the objectives presented in 1.4 and 1.5. For this, 3 components are needed: the virtualization of the UAVs, the simulation of the location probability map of polluting sources and the respective Location and Tracking algorithms. Figure 3.1 shows the main components in a flowchart.

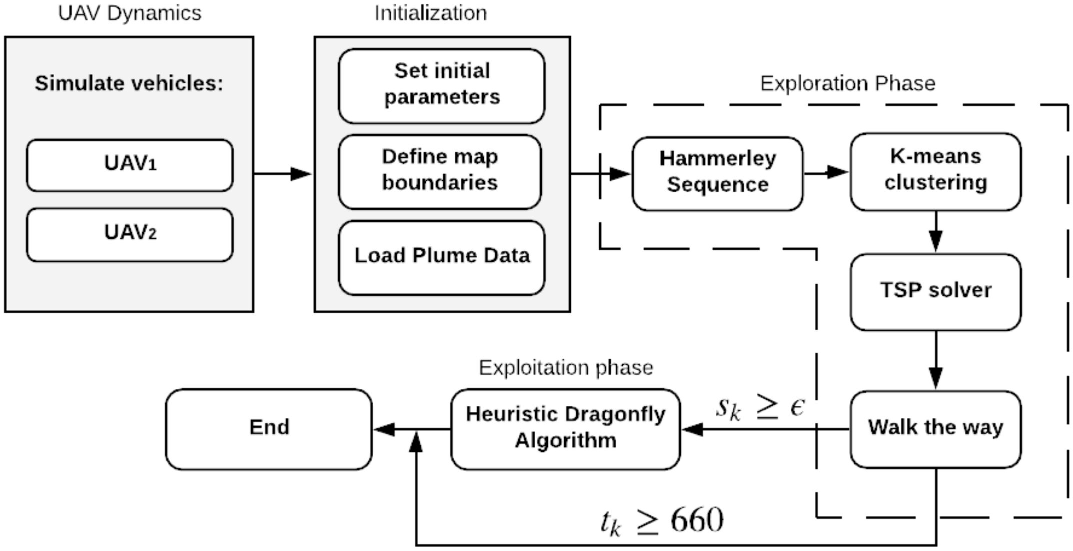


Figure 3.1: Project Methodology

3.1 Virtual UAVs

Ardupilot is an open-source autopilot software based on the Arduino Mega board. With this is possible to control land vehicles, fixed-wing airplanes, airplanes with rotors, helicopters, boats, etc... Ardupilot has a simulator that allows you to run the navigation algorithms without having a physical vehicle. The simulator provides the sensor data from a flight dynamics model in a flight simulator. MAVLINK navigation commands are sent using TCP/IP Sockets on the local network. No specific models of any contamination concentration sensor are used for the simulation. It is assumed that the UAV is equipped with a sensor with sufficient sensitivity to measure contaminants. There are several models that are precompiled. Some specifications of the selected model are presented in Table 3.1. The corresponding airplane is shown in figure 3.2.

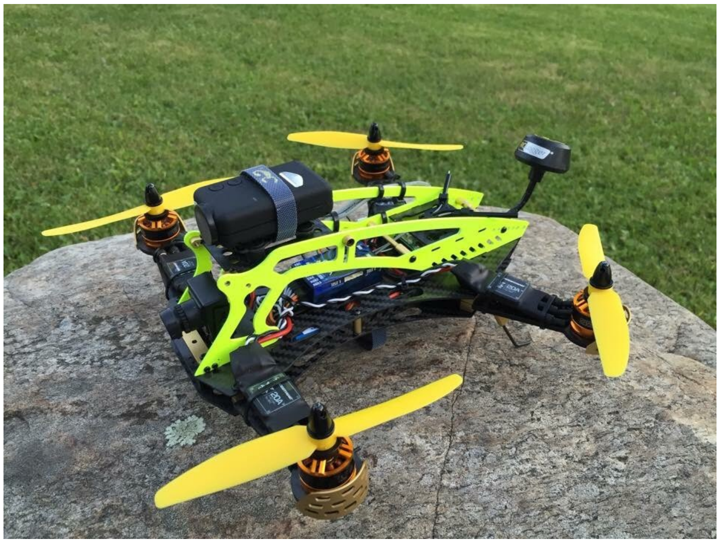


Figure 3.2: The simulated UAV is the Scorpion 280-RATE model

Table 3.1: Parameters of the simulated multicopter

SITL Copter Specifications	
The simulated QUAD	Scorpion 280-RATE
Communication Protocol	MAVLINK
Fly controller	ArduPilot
Connection	Socket TCP/IP
Overall Length	224 mm
Overall height	155 mm
Net Weight	238 g
Diagonal (Motor to Motor)	280 mm
Vertical frame width	41 mm
Motor available size	M-22 ∼M-25

3.2 Probabilistic Map and Plume simulation

The work of the contaminant map depends on the measures of contaminant concentration and the search area. This section will break down the work done to obtain the contaminant map, which is dependent on the measures taken by the virtual UAVs.

3.2.1 Plume Simulation

One of the assumptions mentioned in section 4.1, is that this work considers SO_2 as the air pollutant. Sulfur dioxide is a colorless gas produced by burning coal in power plants or petroleum-based products. This gas causes acid rain. In the air it forms sulfates that are part of PM10 and PM2.5.

The first step to simulate a plume is to calculate the diffusion coefficient. For this, the molecular volumes of diffusion ($\sum v$) for air and SO_2 are required. These values are 19.7 and 41.8 respectively. Using the equation 2.22, the diffusion coefficient of SO_2 in air is calculated as $D = 0.4003e - 3$. Considering an advection coefficient of 0.5 (wind) on the x-axis, the resulting polluting plume, after 20 minutes of emission, is shown in the figure 3.3:

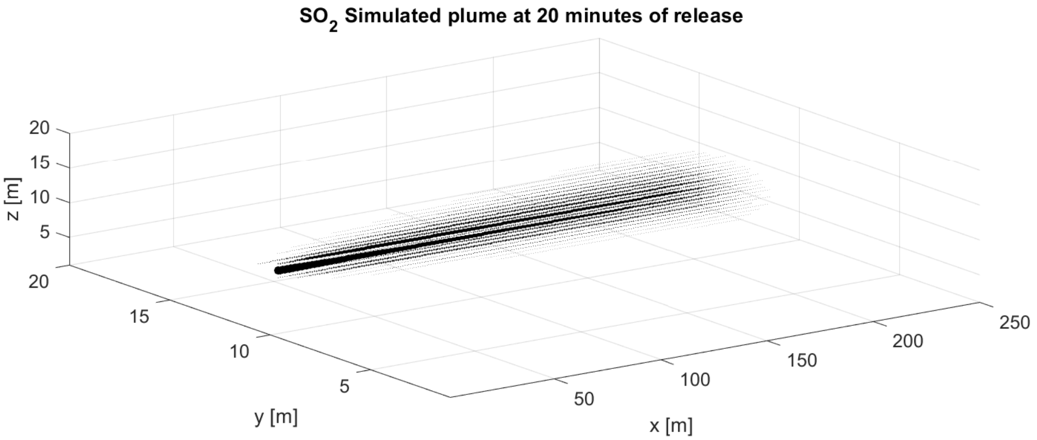


Figure 3.3: Plume Shape after 20 minutes of release

The simulation of polluting plume in real-time will need many computational resources, so instead of doing that, an offline simulation is performed and data is saved in memory. Instead, it will be calculated off-line and then loaded into memory. This will reduce the computational cost of the simulation. The plume will be simulated in a space of 250x20x20. The concentration of each point in space is calculated with a separation of 1 meter in all axis. The source shall be located at point $X_s = (5, 10, 10)$. Each plume space state is stored, from 0 until 1200 seconds. In order to measure the concentration of pollutants, the location of the UAV in GPS coordinates will be mapped to the plume simulation space.

3.2.2 Plume Mapping

The plume is simulated in a rectangular prism. The SO_2 concentration level is maintained in each cubic metre of the prism, i.e. it is discretized. The UAVs fly at a height of 11 and 13 meters, respectively, so the 2D map will refer to the plane z=12 in the space of the plume. To calculate coordinate matching we need the lines that define the search area, then:

Let (Lon_0, Lat_0) the coordinates for the point $(0, 0, 12)$ in the plume space and $R = 6378137$ the radius of spherical earth. The (Lat_f, Lon_f) coordinates for the point $(250, 20, 12)$ are:

$$Lat_f = Lat_0 + \frac{Y \cdot 180}{R \cdot \pi} \tag{3.1}$$

$$Lon_f = Lon_0 + \frac{X \cdot 180}{\pi \cdot R \cdot cos(\pi \cdot Lat_0/180)} \tag{3.2}$$

Where $X = 250$ and $Y = 20$. The condition for the UAV to take a SO_2 concentration sample is:

$$UAV_{Lat} \in [Lat_0, Lat_f] \; and \; UAV_{Lon} \in [Lat_0, Lat_f] \tag{3.3}$$

The correspondence between the GPS coordinates of the UAV and the coordinates of the plume is shown below:

$$
\begin{aligned}
\Delta Lat =& UAV_{Lat} - Lat_0 \\
\Delta Lon =& UAV_{Lon} - Lon_0 \\
x_{index} =& \lfloor R \cdot (\Delta Lon \cdot \pi/180) \rfloor && (3.4) \\
y_{index} =& \lfloor R \cdot (\Delta Lat \cdot \pi/180) \rfloor && (3.5)
\end{aligned}
$$

With this relationship, it is already possible to obtain the concentration values of SO_2, to perform the search. The next step in the project is to obtain the probability map.

3.2.3 Probabilistic Map

The objective of the navigation algorithm is to move every UAV towards the highest level of pollution. For this, the system needs some criteria to select the next position to go. The probability map will be part of this criterion, in the proposed strategy. From this point, we will call ”heuristic” to any additional information that helps us to select a new UAVs position. Section 2.8 shows the theory needed to generate a map in a specific area. In this section, suitable changes will be made to use the map as a heuristic of the search algorithm and the location algorithm. In addition, the map delimitation algorithm is presented, which depends on the UAV take-off location.

3.2.4 Map delimiters

In order to compare the answers to each search strategy, it is necessary to have specific starting points. This is so that the experiments are on an equal footing. Due this, the probability map must be in the same place in all experiments. In addition, all maps must have the same values of: N, M as the number of squares on the x and y axis and D_x, D_y as the square length and width.

However, defining n_0, m_0 as the initial UAV cell position in the map and (UAV_{lat}, UAV_{lon}) as the current position of the UAV, the distances towards every map border are:

$$
\begin{aligned}
dist2Lon_U =& (N - n_0 - 0.5) \cdot D_x/N \\
dist2Lat_U =& (M - m_0 - 0.5) \cdot D_y/M \\
dist2Lon_L =& (n + 0.5) \cdot D_x/N \\
dist2Lat_L =& (M + 0.5) \cdot D_y/M && (3.6)
\end{aligned}
$$

The value of 0.5 is added as we assume that the UAV takes off from the center of the Cell (n_0, m_0). Then, using equations 3.1 and 3.2, the boundaries of the map are as in figure 3.4.

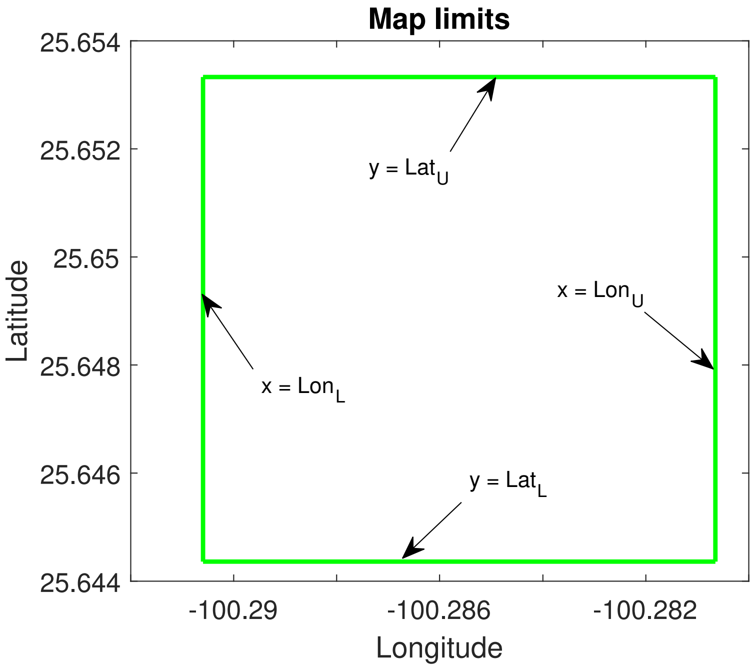


Figure 3.4: Map boundaries established at the time of take-off

3.2.5 Cell subdivisions

Once the search area is delimited, the map is subdivided into smaller squares. The map will be divided in a similar way to the one shown in figure 2.5. The main difference will be the reference shape of each cell. The way to do this will be as if it were a 2-dimensional matrix, that is, we will not use equations 2.26 and 2.26 proposed by [23].

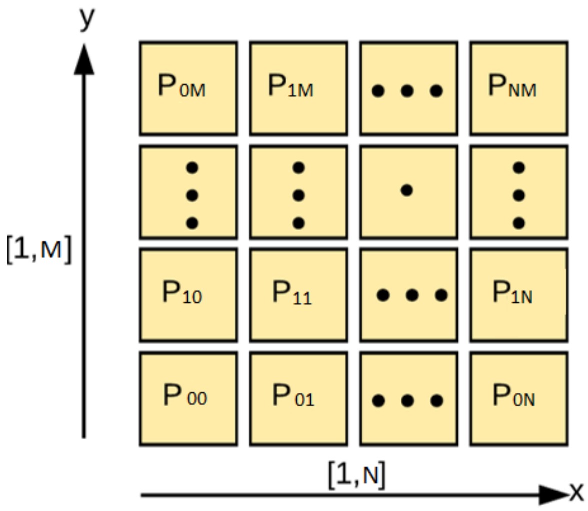


Figure 3.5: How to reference the cells in the probabilistic map

The location of each cell center, on the real map, is calculated as:

$$Lat(m) = Lat_L + \frac{Lat_U - Lat_L}{M} \cdot \left(m + \frac{1}{2}\right) \tag{3.7}$$

$$Lon(n) = Lon_L + \frac{Lon_U - Lon_L}{N} \cdot \left(n + \frac{1}{2}\right) \tag{3.8}$$

With this system, the probability map also allows us to have fixed UAV positions within the search area.

3.2.6 Bayesian-Based Map

The work of [23] presents a probability vector for the air pollutant source location. In this section, simplifications of equations 2.28, 2.29 and 2.30 will be used to obtain a probability matrix. This matrix is directly related to the equations 2.31 and 2.32, but in the reference system shown in the section 3.2.5. The equations for calculating the probability matrix, with respect to those shown in section 2.8.2, are:

$$S_{ij}(t_k) = \frac{e^{-\frac{(|n_j - n_i| - v_x)^2}{2t_k\sigma_x^2}} e^{-\frac{(|m_j - m_i| - v_y)^2}{2t_k\sigma_y^2}}}{2\pi t_k \sigma_x \sigma_y} L_x L_Y \tag{3.9}$$

Note the absolute value in the exponential. This considers the closeness of the cells, but not their position in the matrix. The other equations are:

$$\beta_{ij}(t_k) = \frac{S_{ij}(t_{k-1}) + S_{ij}(t_k)}{k} \tag{3.10}$$

$$\gamma_{ij}(t_k) = (1 - S_{ij}(t_{k-1}))(1 - S_{ij}(t_k)) \tag{3.11}$$

For the calculation of the probability matrix, based on equations 2.31 and 2.32:
If there's a detection in C_j:

$$\alpha_{ij}(t_k) = NM\alpha_{ij}(t_{k-1})\beta_{ij}(t_k) \tag{3.12}$$

If there's not a detection in C_j:

$$\alpha_{ij}(t_k) = \frac{NM\alpha_i(t_{k-1})\gamma_{ij}(t_k)}{\sum_{M}^{i=1} \gamma_{ij}(t_k)} \tag{3.13}$$

Where i is for the source location and j is for the current UAV location. With the formulas shown above it is possible to update the probability matrix (or map), given the condition that a high level of SO_2 concentration is detected or not. In order to obtain this measure, it is necessary an UAV Navigation system. This system will be explained in the next section.

3.3 Navigation System

The navigation system guides the UAV to the desired locations. In this paper we will consider the use of 2 UAVs. The UAVs do not work as a single unit, i.e. they do not move in a coordinated way. Figure 3.6 shows the most basic navigation algorithm.

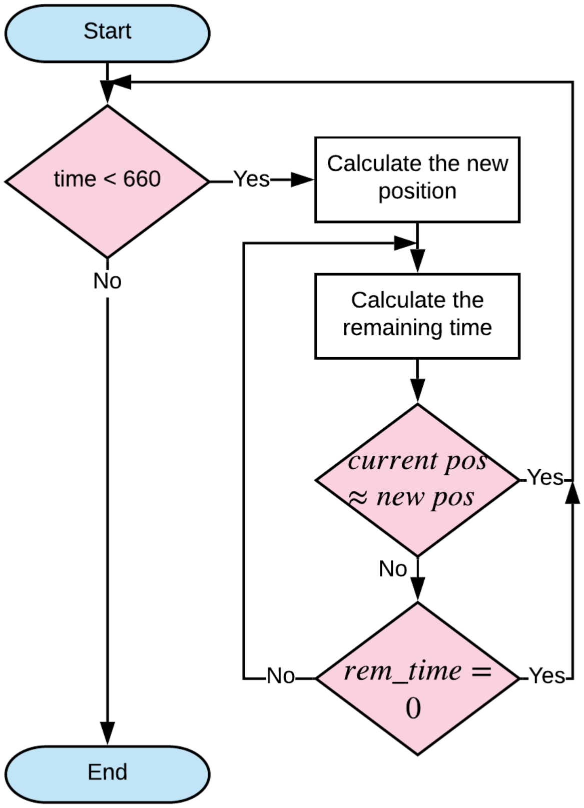


Figure 3.6: Flowchart of the basic Navigation System

The rem_time variable is the time for move the UAV until the next point at $8\ m/s$. This variable is necessary because the condition to ensure that you reached the target position may fail. In order to say that the current position is approximately the target position, it is considered that at least 90% of the calculated path has been travelled. This measure works very well over long distances. If the target point is close enough, the condition will not be met, so the algorithm will assume that the target was never reached. This time condition causes the program not to stop if this situation occurs. Remaining Path is computed with the Haversine formula:

$$\begin{aligned} a &= sin^2(\Delta Lat/2) + cos(Lat_1) \cdot cos(Lat_2) \cdot sin^2(\Delta Long/2) \\ c &= 2 \cdot atan\left(\sqrt{\frac{a}{1-a}}\right) \\ d &= R \cdot c \end{aligned} \tag{3.14}$$

3 approaches are considered as a basis for the development of the proposed navigation system:

1. Greedy Search
2. Cooperative Greedy Search
3. Dragonfly Algorithm

Each of these approaches are explained in the following sections.

3.3.1 Greedy Search as a Navigation Algorithm

As explained in Section 2.3, a greedy quest is based on following the path that maximizes my profit. Some of the advantages of implementing this approach are:

- Its implementation is easy.
- It doesn't usually require much computational expense.
- It is a fast algorithm it does not require much time to converge to a solution.
- The answer of a greedy search can be used as heuristics for any other optimization algorithm.

The disadvantages of this approach are:

- The decisions it makes are optimal but at the local level.
- Due to above, in many cases they do not find a global optimum.
- A formal demonstration that the algorithm found an optimal solution is necessary.

Taking as heuristics of the algorithm the probability map shown in section 3.2.6, the next position in the navigation is given by:

$$\vec{X} = (Lon(n), Lat(m)) \mid \forall o \in [0, N] \wedge \forall p \in [0, M] : \alpha(o,p) \leq \alpha(n,m) \tag{3.15}$$

Updated probability map, with a greedy algorithm and the characteristics develop in section 3.5, is showed in the figure 3.7. Each cell represent the Probability of locating the source in the cell. Blue areas represent low probability and yellows areas high probability.

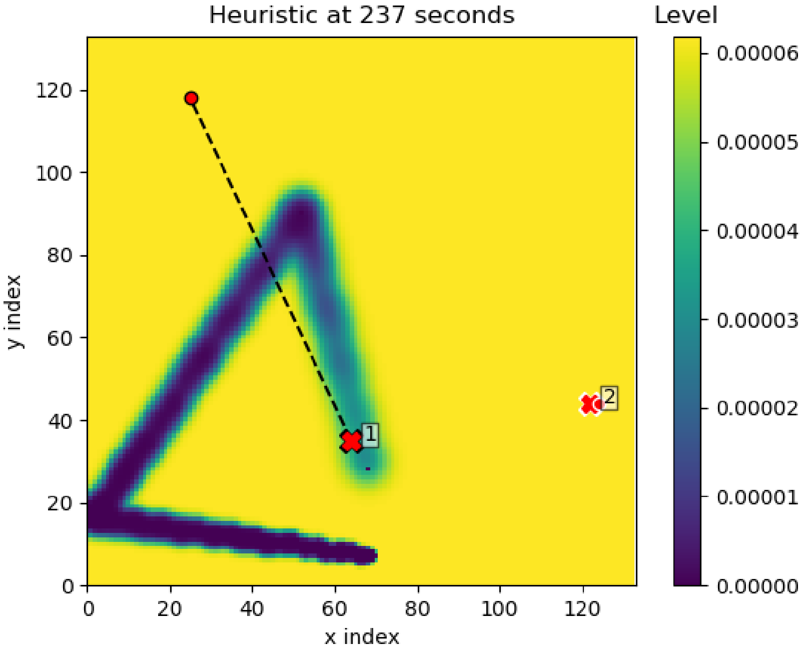


Figure 3.7: Map updated by a Greedy Search algorithm

The flowchart to calculate the new position is:

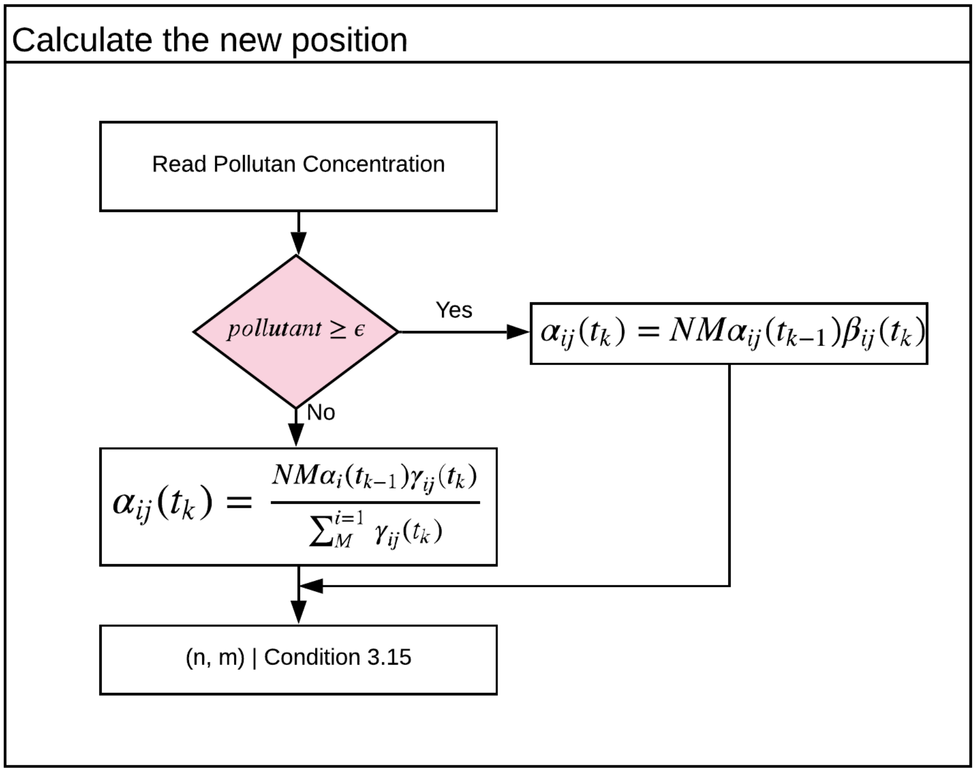

Calculate the new position
Read Pollutan Concentration
$pollutant \geq \epsilon$
Yes
No
$\alpha_{ij}(t_k) = NM\alpha_{ij}(t_{k-1})\beta_{ij}(t_k)$
$\alpha_{ij}(t_k) = \frac{NM\alpha_i(t_{k-1})\gamma_{ij}(t_k)}{\sum_M^{i=1} \gamma_{ij}(t_k)}$
(n, m) | Condition 3.15

Figure 3.8: Flowchart to calculate a new position on the probabilistic map

3.3.2 Cooperative Greedy Search

In the previous section, we show in the figure 3.7 how 2 UAVs are used to find a pollution source. The focus was on independent navigation, i.e. the status of the other UAV didn't influence one's own decisions. In this section the cooperative interaction of 2 UAVs is considered. The system's heuristic is the probability map of section 3.2.6. If this resource can be modified by any UAV, it is obvious that each UAV influences the other. The advantages of this approach are:

- In theory, the probability matrix is updated at twice speed.
- The response is less likely to converge to a local maximum or minimum.

As the map can be modified at any time by any UAV, there may exist conflicts. The following section discusses this issue.

Race Condition

The probability map is expected to be modified by any UAV. There is no priority of one UAV or another. This is useful because any UAV can find a detection at any time. The probability map is instantiated in one UAV and backed in the other, that is, updates will be made in only one of the UAVs. To avoid the race condition a mutex condition is issued,, thus restricting the modification of the probabilistic map. A mutex in ON allows modify or read the map. The mutex in OFF state will stop the navigation control (not the flight control) of the requesting UAV. This means that a new position is not calculated until the map is updated. As the probability matrix is the heuristic of the system, and the greedy search algorithm doesn't use any more information, the new position of a UAV is calculated with 3.15. An example of the probability map of this approach can be seen in Figure 3.9. Each cell represent the Probability of locating the source in the cell. Blue areas represent low probability and yellows areas high probability.

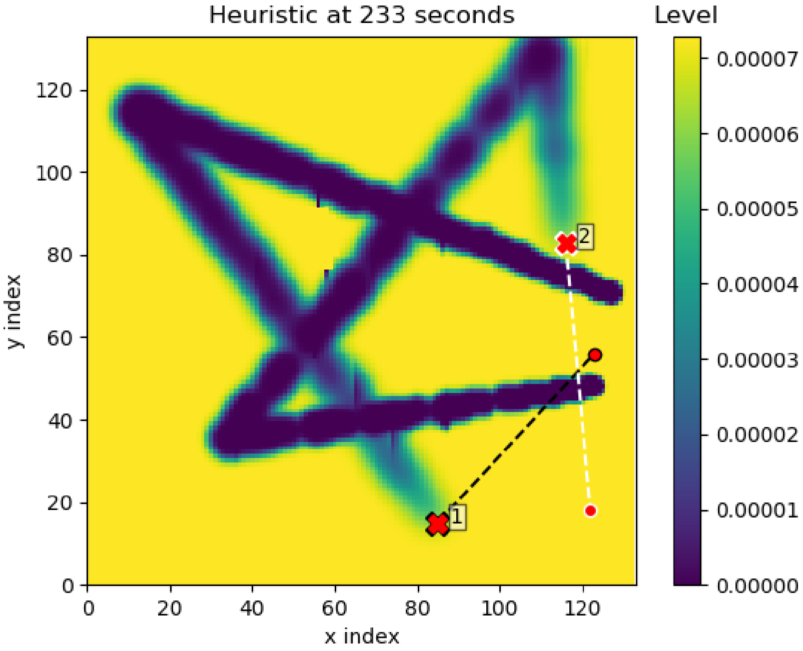


Figure 3.9: Cooperative update of probability map.

Although the behavior of this algorithm has a good performance, it's not efficient. Figure 3.10 shows the path of 2 UAVs with the greedy search algorithm. The area of interest is the ellipse. On several occasions, UAVs explore areas that were explored before. Although this is not necessarily bad, it is much better to explore new areas of the map.

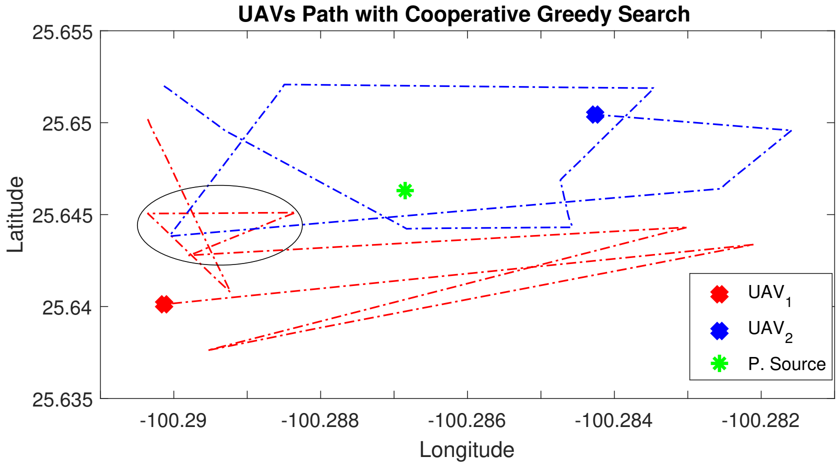


Figure 3.10: The enclosed area is visited many times by UAVs.

To avoid this behavior, the following approaches use a 2D low discrepancy sequence.

3.3.3 Dragonfly Navigation Algorithm

In section 2.4 the Dragonfly Algorithm was introduced. The original article ([22]) aims to demonstrate the effectiveness of the algorithm by using it with various functions. The answers of the maximums found are compared with the answers of other optimizers. In the end the efficiency of this algorithm is justified.
The algorithm is inspired by the behavior of flying insects. Therefore, a navigation control using this algorithm should perform well. Equations 2.1, 2.2, 2.3, 2.4 and 2.6 are the components of the final step vector (see figure 3.11). To calculate each component the location information of the other UAV and its step vector are essential.

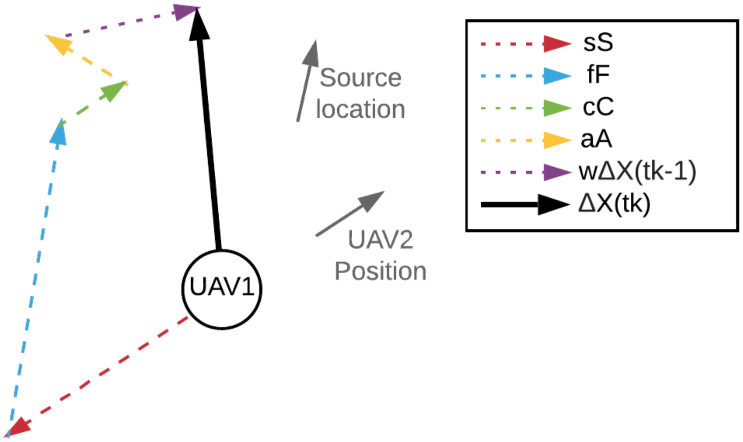


Figure 3.11: The step vector is the sum of all components (Eq. 2.6).

In the previous figure: sS is the separation component, fF is the food attraction component,

cC is the cohesion component, aA is the alignment component, $w\ DeltaX$ It is a component of inertia and $\ DeltaX$ is the vector step.
There are 2 steps necessary to search and trace the source:

- Exploration
- Exploitation

The next sections will discuss this stages.

3.3.4 Exploration stage

The goal of this stage is to navigate the search area as efficiently as possible. A Scheme of Methods used by the algorithms is shown in the figure 3.12. Since no detection has yet been made at this stage, it is advisable that the food attraction component does not influence the route. There is an inconvenient to using the dragonfly algorithm in the exploration stage: The components of separation and cohesion are opposite. Under certain conditions these components can cause a loop of oscillations of the positions of the UAVs. In order to avoid this behavior at this stage it was decided to use low discrepancy sequences. Specifically, a Hammesley sequence is used.

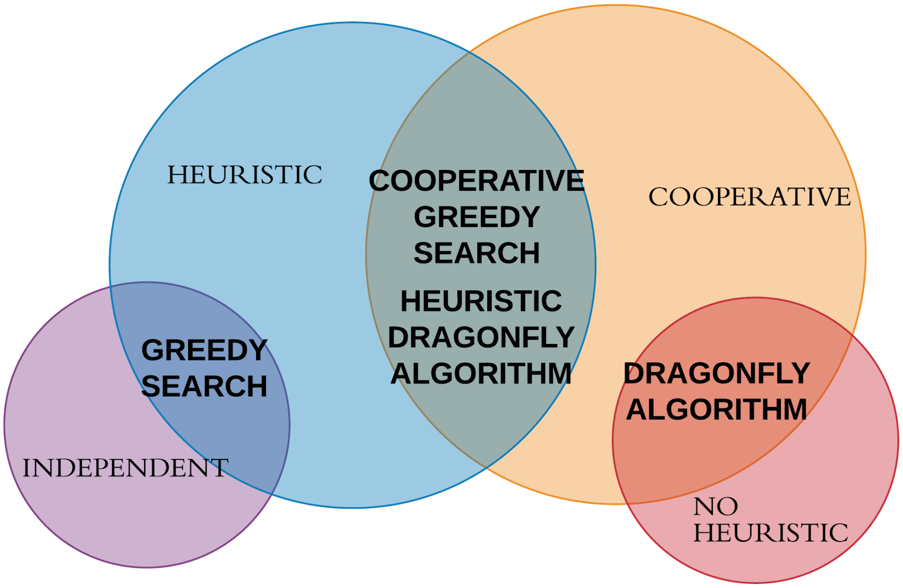


Figure 3.12: Scheme of Methods used by the algorithms

Hammersley Sequences

The Hammersley sequences (look section 2.5) have the following characteristics:

- The points generated are quasi-random.
- The sequences bases must be co-prime numbers.

- The first sequence number be transformed into Hammersley is random.
- The bases of the sequence are randomly selected prime numbers.

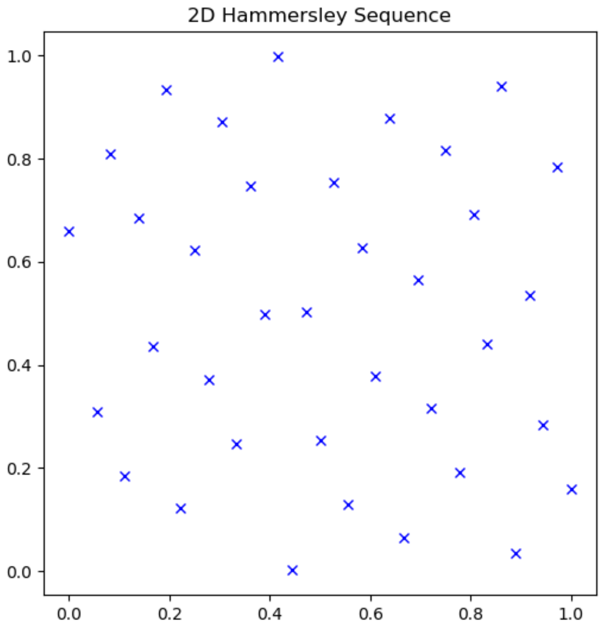


Figure 3.13: 35 points from a 2D Hammersley sequence.

The Hammersley sequence points have a range of 0 to 1, but it is easy to accommodate them on the probability map. Now, it's necessary to select which points will run with each UAV. A simple method is dividing the map into 2 and assign each group to an UAV. This method isn't always good. Random nature can generate disparate groups. To solve this problem the k-means grouping algorithm was used.

Clustering and Traveler Salesman Problem

Grouping using the k-means technique was explained in section 2.6. In this project the centroids will be initialized randomly. Once assigned the points that each UAV will travel, a technique is needed that allows them to travel efficiently. This is a case of the Traveling Salesman Problem (TSP) [10]. The original problem is: there is a salesperson who must visit a certain number of cities. The distance between each city is known. The traveler must visit each city once following the shortest route.

A Greedy Search was used to solve this problem. As an example, the Hammersley points in Figure 3.13 were grouped and sorted to create the path for each UAV. The assignment can be seen in figure 6.

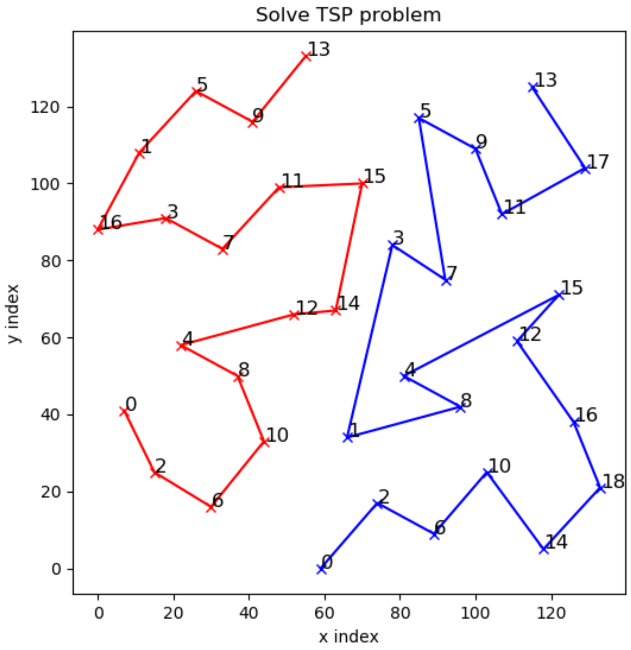


Figure 3.14: Solution of 2 Travel Salesman Problems. Each number is the Hammersley position in each sequence.

Now, the scanning sequence of each UAV is complete. This operation will continue until a high contaminant detection is performed, or until the simulation time is over. A Hammersley sequence of 35 points was chosen because the time needed to traverse all points is usually 10 minutes. Figure 3.15 shows the probability map updated with the described methods. Areas of yellow represent a high probability of locating a polluting source in the cell. Areas of blue represent a low probability of locating a polluting source in the cell.

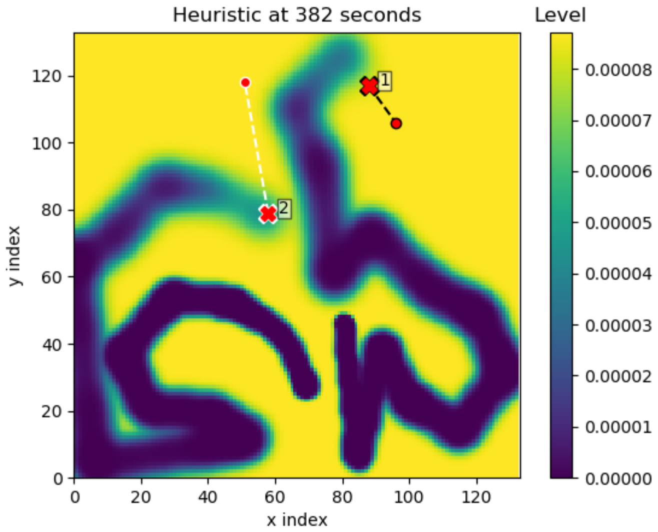


Figure 3.15: Updated map using Hammersley sequences, clustering and TSP.

3.3.5 Exploitation Stage

The start of this phase may be due to 2 situations: The Hammersley sequence was either completely run through or a high level of contaminant concentration was detected. For each case different values of s, a, c, f and w are taken:

- At the beginning it will be required that the UAVs are separated from each other, then $s = 0.7$.
- Since there was no detection $f = 0$.
- The alignment component can vary, in this case a random value between 0 and 1 will be chosen.
- To maintain the separation, the cohesion component must be low, then $c = 0.1$.
- The value of w must be low so that the UAV is mostly influenced by the other components, then $w = 0.3$.

If in this phase a high concentration of pollutants was detected in the air, then:

- We want that the UAVs to tend to the food point, then $f = 0.7$.
- UAVs should tend to get together, then $c = 0.5$.
- UAVs must not oscillate or stop at a point, they must be a certain level of separation, then $s = 0.3$.
- The alignment component shouldn't be influential at this stage, then $a = 0.1$.

- To maintain a heading, the inertia must be high, but not greater than the food influence, then $w = 5 \cdot Random(0, 1)$.

The functioning of this algorithm is strongly influenced by the locations of each UAV. In figure 3.16 the flowchart of this algorithm can be described for any of the 2 states explained above.

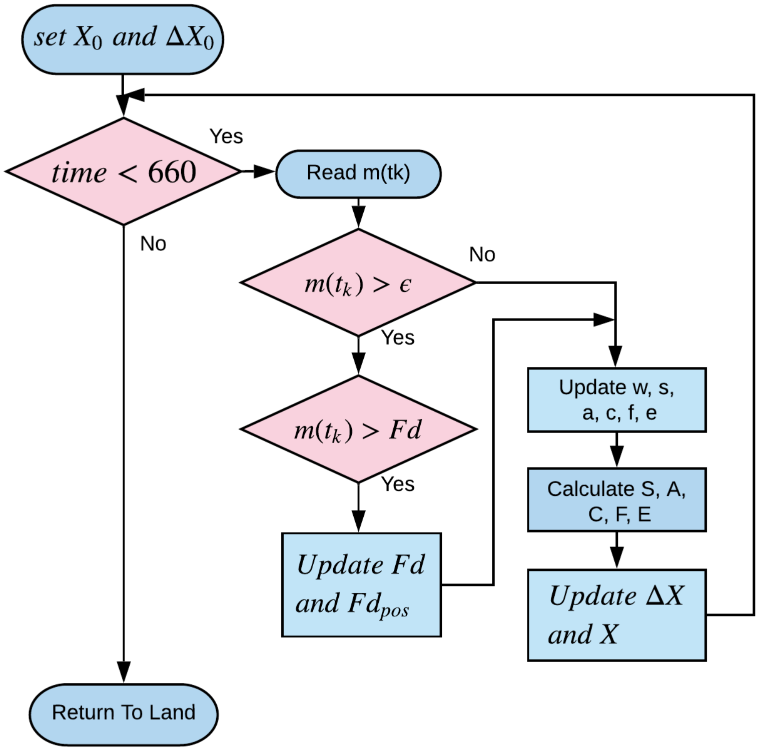


Figure 3.16: Flowchart for Dragonfly Algorithm

3.4 Heuristic Dragonfly

The approach of this section relates the heuristics calculated in section 3.2.6 to the dragonfly algorithm shown in section 2.4. As mentioned before, 4 components of the dragonfly algorithm are weighted and adding them gives the new UAV location. The behavior of each individual will depend very much on the weighting parameters. The selected parameters described in section 3.3.4 were manually tuned. In many configurations it was the case that the UAVs were close to the source, but before arriving they were far from it. In other cases they began to change location and the centroid displacement was almost zero. The final parameters make the UAVs select positions very close to the source. If a higher counting value is found in one of these locations, there will be a displacement of the centroid. If not, the UAVs will move from one side to the other until they find it. The inconvenient with this process is that UAVs will repeat the same positions close to the food many times over. To avoid this, a modified version of the algorithm uses the probability map. With this probability map it will be less likely that UAVs will repeat a position. The implementation of this system is nothing more than adding a new component to the dragonfly algorithm. Then, the heuristic component will be:

$$H_i = (Lat(n), Lon(m)) - X \tag{3.16}$$

And it will be weighted by $h = 0.3$. Figure 3.17 shows a flow chart of the algorithm.

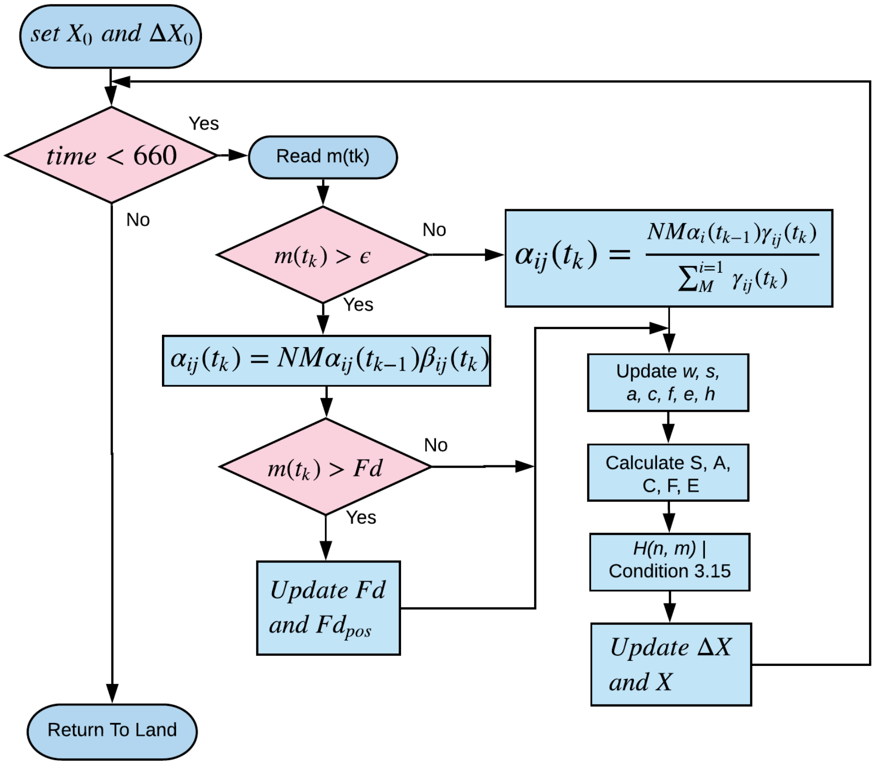

set X_0 and ΔX_0
time < 660
Yes
No
Read m(tk)
$m(t_k) > \epsilon$
No
Yes
$\alpha_{ij}(t_k) = \frac{NM\alpha_i(t_{k-1})\gamma_{ij}(t_k)}{\sum_M^{i=1} \gamma_{ij}(t_k)}$
$\alpha_{ij}(t_k) = NM\alpha_{ij}(t_{k-1})\beta_{ij}(t_k)$
$m(t_k) > Fd$
No
Yes
Update Fd and Fd_{pos}
Update w, s, a, c, f, e, h
Calculate S, A, C, F, E
H(n, m) | Condition 3.15
Update ΔX and X
Return To Land

Figure 3.17: Dragonfly Flow Chart that shows the Heuristic component

Chapter 4

Simulation and Results

This section presents results of each approach to searching and locating air pollution sources. Real-time simulations aim to approximate the behavior of a platform with Arducopter and Python in a search area. There are 3 takeoff zones for the UAVs: at 250 ($Z1$), 450 ($Z2$) and more than 450 ($Z3$) meters from the pollutant source. Figure 4.1 shows the locations of drones taken into account.

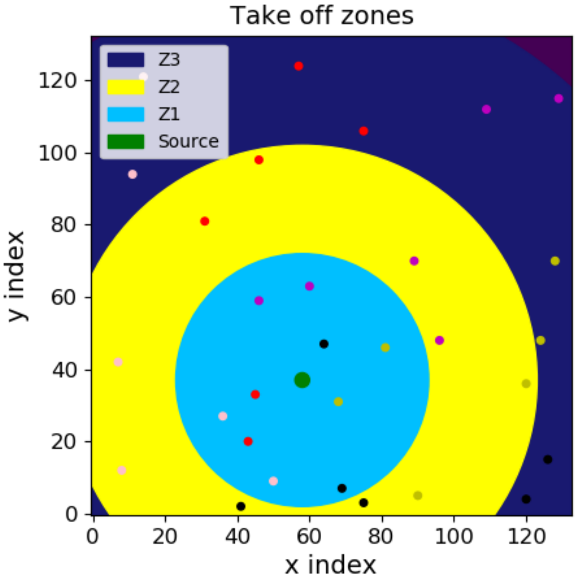


Figure 4.1: UAV take-off Zones. Each pair of dots of the same color symbolizes UAVs location for take-off.

Considering that the feather has a width of 14 m, a length of 160 m, at a height of 13 m after 20 minutes of contaminant leakage, the percentage of the feather with respect to the search area is:

$$\frac{14 \cdot 160}{100 \cdot 100} \cdot 100\% = 2.24\%$$

The relation between the area of covered by the plume and the search area is shown graphically in figure 4.2.

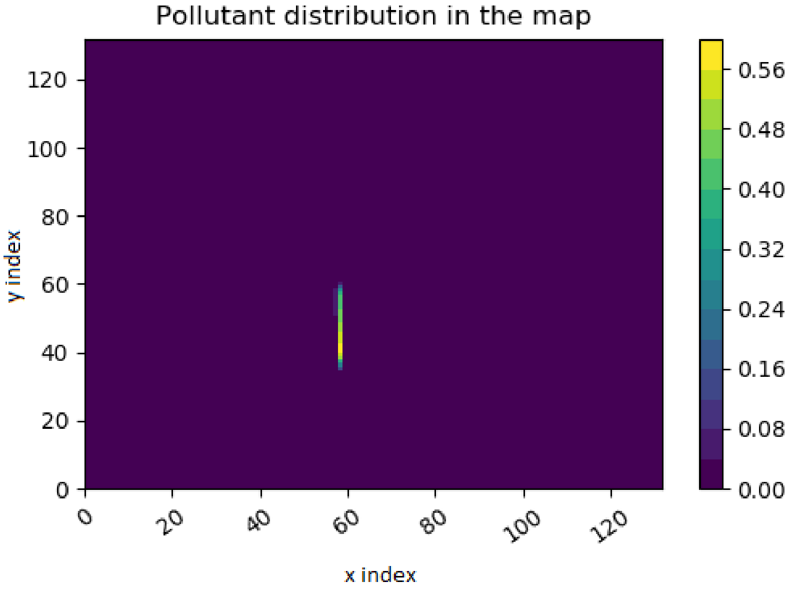


Figure 4.2: Plume location on the map

To better observe the feather figure 4.3 shows an area of 167x300 meters around it.

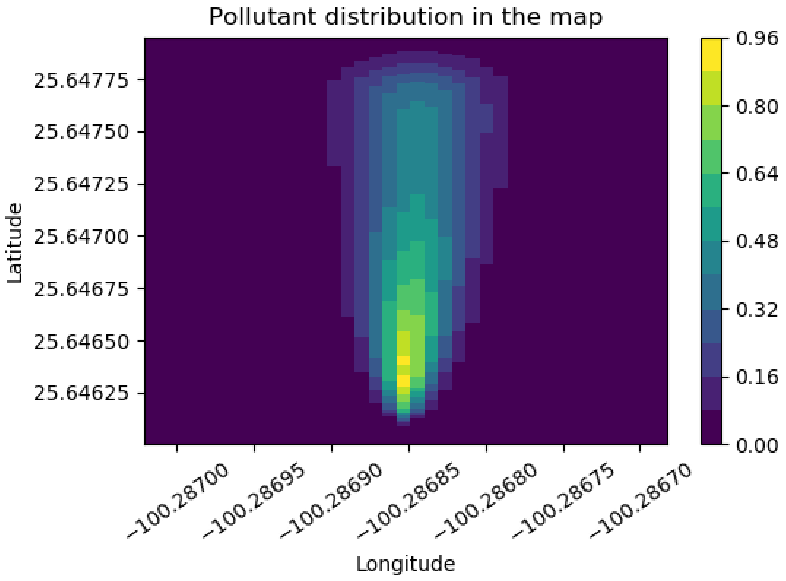


Figure 4.3: Plume simulated in an area of $167x300\ meters^2$

4.1 Parameters and Assumptions

The simulation requires some initial parameters to run. In addition, it is necessary to establish conditions under which the simulation can be executed on a real platform. The parameters and assumptions of the project are shown below:

1. The threshold for high levels of pollutants was obtained from the Mexican standard NOM-038-ECOL-1993, for SO_2.
2. The probabilistic map size is 1 km^2, divided in 133×133 squares.
3. The UAV ground speed is 8 m/sec.
4. The probabilistic map needs the dispersion parameters in x and y axis. This values are $\sigma_x = 0.3$ and $\sigma_y = 0.3$
5. As mentioned in section 2.7.5, the wind remains in constant direction during the simulation period, at $1.3\ m/s$. Data from an ultrasonic anemometer were used to justify this assumption. Measurements were taken every 10 minutes. A statistical test was performed with the difference of the samples. A T test returned a value of p=0.9922, which indicates that the null hypothesis (mean equals zero) cannot be rejected.
6. There are no obstacles for UAVs.
7. UAVs cannot collide due both are flying in different altitudes.
8. The sensors give good measurements, even with the turbulent air flow generated by the UAVs propellers [26].
9. The GPS gives good measurements.
10. UAVs antennas have a range of at least 1 km.
11. The source of contaminants has been present for an indeterminate time period.
12. The battery allows 12-minutes of flight.

4.2 Detections and no Detections

This section shows the percentage of experiments which resulted in a detection. In section 4.5 the interpretation of the figures is shown. In order to carry out the subsequent analysis, it is necessary to discard the results in which there was no detection. This is because, as explained in the 3.3.5 section, the Dragonfly algorithm will not work until detection occurs.

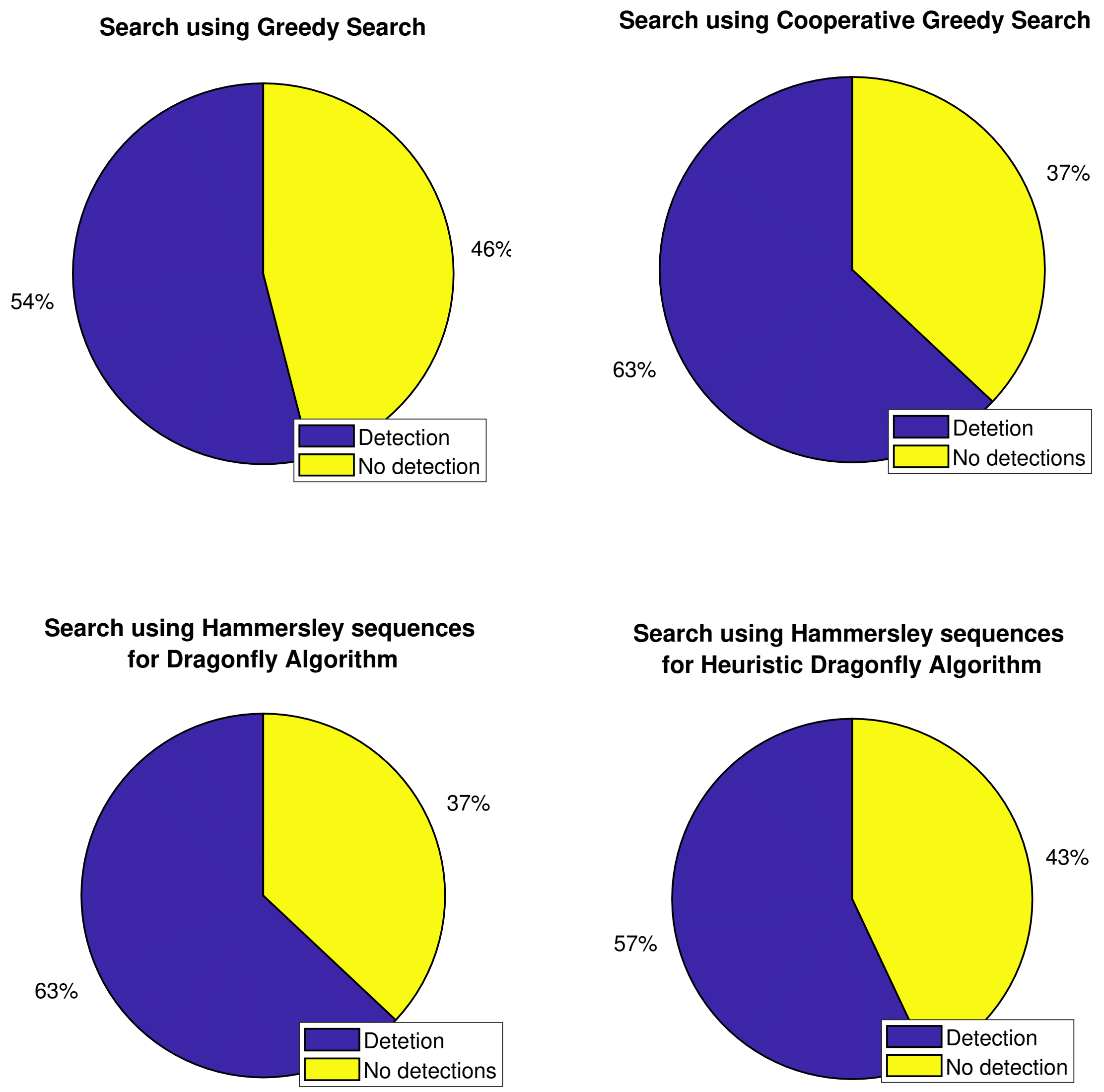


Figure 4.4: Percentages of detection of high levels of concentration of pollutants in the air

4.3 Air Pollutant Source Proximity

This section shows the distance between each UAV and the pollutant source. These results serve to discuss the behavior of each approach.

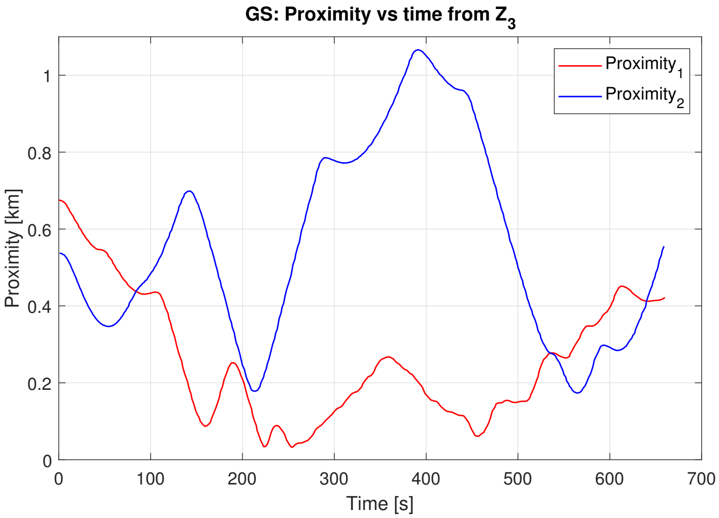


Figure 4.5: Proximity vs. time of the Greedy Search Algorithm

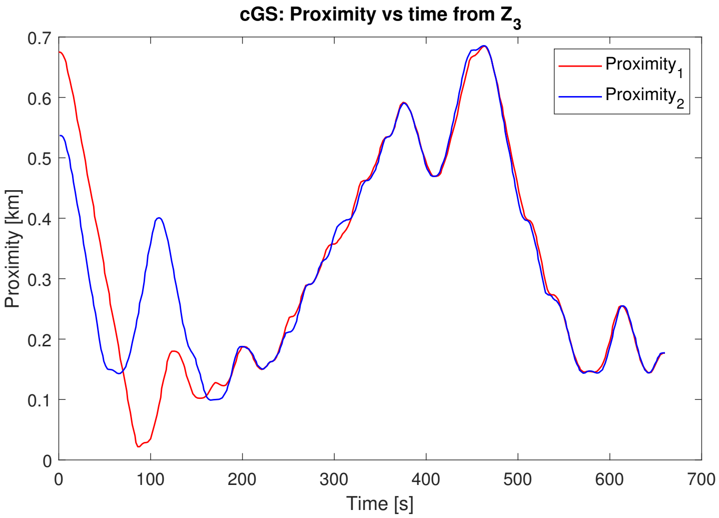


Figure 4.6: Proximity vs. time of the Cooperative Greedy Search Algorithm

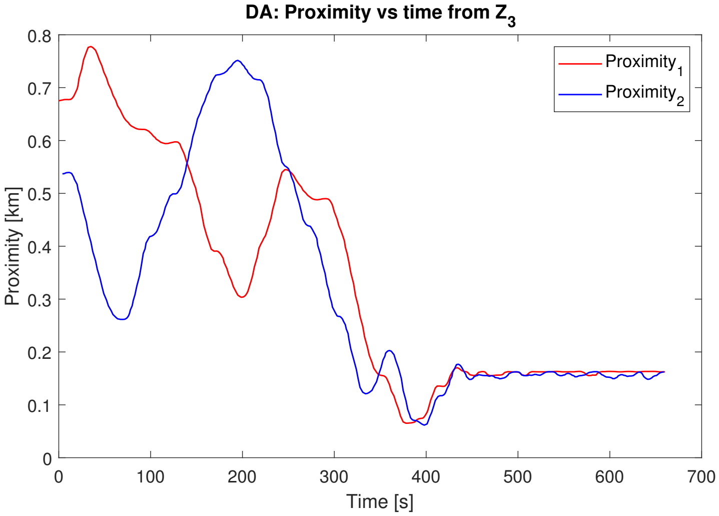


Figure 4.7: Proximity vs. time of the Dragonfly Algorithm

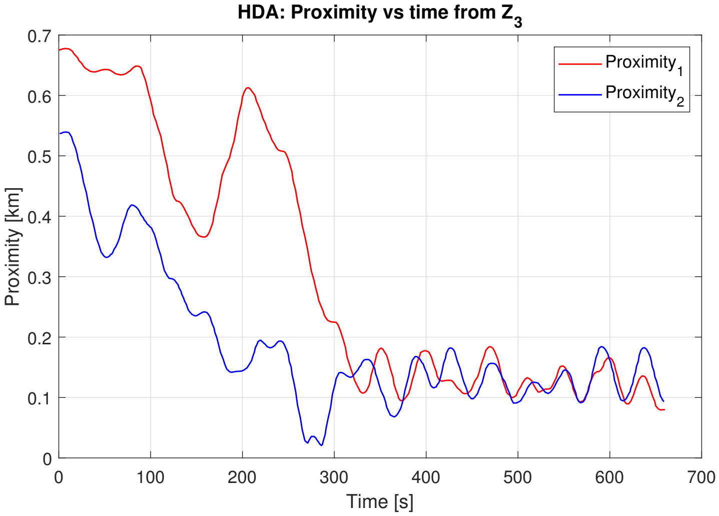


Figure 4.8: Proximity vs. time of the Heuristic Dragonfly Algorithm

4.4 Distribution of Results

This section presents data summarized by boxplots. Boxplots are a statistical tool that allows us to analyses results distribution. The analyzed characteristics are:

- Point closest to the measured pollutant source.
- Highest level of measured pollutant concentration.
- Time elapsed until the highest detected contaminant level is measured.
- UAVs path to measure the highest contaminant level detected.

In the figures, the acronyms are: GS for the Greedy Search algorithm, cGS for the cooperative Greedy Search Algorithm, DA is for the Dragonfly Algorithm and HDA is for the Heuristic Dragonfly Algorithm. The shading cells indicates the best statistical value.

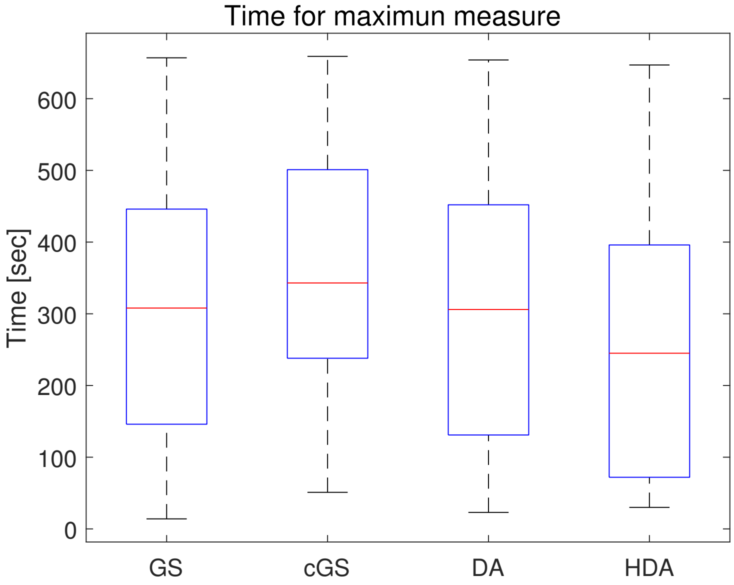


Figure 4.9: Boxplot for time dispersion of results

Table 4.1: Statistical values for time responses

	GS	cGS	DA	HDA
Median	308	343	306	245
σ	183	172	181	176
Mean	228	312	242	183
Q1	143	238	129	72
Q3	452	505	453	403

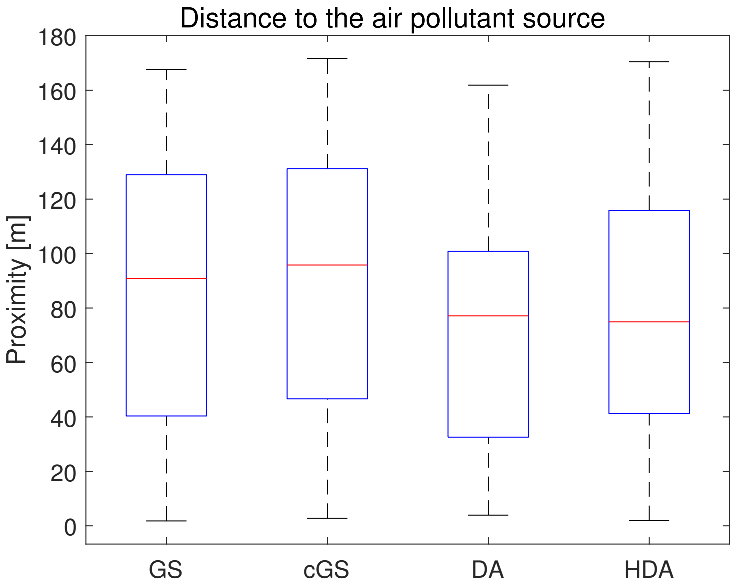


Figure 4.10: Boxplot for proximity dispersion of results

Table 4.2: Statistical values for proximity responses

	GS	cGS	DA	HDA
Median	90	95	77	74
σ	47	47	47	47
Mean	64	71	60	55
Q1	40	46	33	41
Q3	129	131	102	117

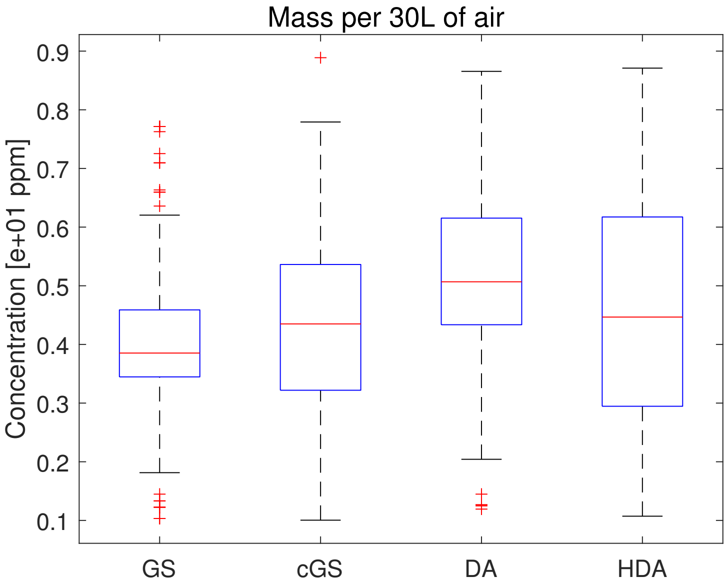


Figure 4.11: Boxplot for air pollutant concentration of results

Table 4.3: Statistical values for concentration dispersion of responses

	GS	cGS	DA	HDA
Median	0.038	0.043	0.050	0.044
σ	0.017	0.017	0.019	0.021
Mean	0.037	0.039	0.047	0.042
Q1	0.034	0.032	0.043	0.029
Q3	0.046	0.054	0.062	0.062

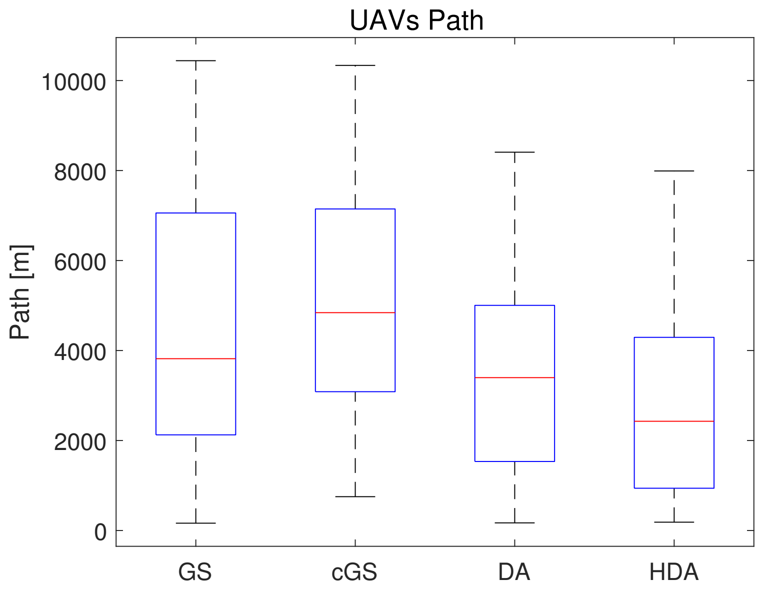


Figure 4.12: Boxplot for path dispersion of results

Table 4.4: Statistical values for path responses

	GS	cGS	DA	HDA
Median	3818	4841	3397	2427
σ	2793	2791	2182	1986
Mean	3274	4336	2787	2020
Q1	2044	3085	1533	934
Q3	7057	7147	5004	4358

Finally, in order to demonstrate that the results are significantly better, a single-factor ANOVA was performed among the data obtained.

Table 4.5: Single-factor ANOVA results

	Time	Concentration	Proximity	Path
p_value	0.0009	0.0005	0.1475	0

4.5 Analysis of results

The proposed strategy have specific characteristics that make it different from other works. A quantitative comparison is not possible, therefore the table 4.6 shows a qualitative comparison between some similar works.

Table 4.6: Characteristics of the proposal against similar works

Characteristics	**Proposal**	**Jatmico, 2007 [14]**	**M. Rossi, 2016 [26]**	**J.J. Cassano, 2016 [4]**	**John, A. A. B., & Dutta, R. 2017 [15]**	**J. R. Bourne, 2019 [3]**
Real time Simulation	YES	NO	Real environment	Real environment	NO	NO
Large search area	YES	NO	NO	YES	NO	NO
Cooperative strategy	YES	YES	Only 1 UAV	YES	YES	YES
Online plume simulation	NO	YES	Real environment	Real environment	YES	YES

Continuing with the discussion of results, the next point to be discussed is the results of figure 4.4. The percentage of detections between the dragonfly algorithm and Greedy Search with cooperation are the same. In one of the methods, Hammersley sequences were used to make the travel of UAVs more efficient. The efficient travel peculiarity of the Hammersley sequences is compared to the speed at which a greedy search travels the map. This is because with Hammersley sequences the UAV must travel over concave routes, as seen in figure 3.14, so now the UAV must rotate in Yaw, before continuing its traveling, reducing its speed and taking more time for the maneuver. On the other hand, in a greedy search, this behavior is not so common. The UAV also must rotate in Yaw to change direction, but it does it less often.

The next point to discuss is the performance of each approach regarding the source proximity. Experiments in which a high level of pollutants (concentration > 0.1 ppm) has been detected were used. The UAVs take-off from the same location. The first figure (4.5) shows the behavior of the greedy search using the probability map. When one of the UAVs detects a high level of contaminants, it changes course to go to the detection point. It stays relatively close trying to detect a new point with a high concentration of contaminants. Since no information is shared, the other UAV continues to search without approaching any specific point.

The behavior of the Cooperative Greedy Search is a bit different. The disadvantage of this approach is that when a detection is made, the UAVs tend to search the same paths, as can be seen in figure 4.5. This is because in the proximity map there will be just a few points with a high probability of being a polluting source.

The advantage of the Dragonfly Algorithm over the approaches is that, once a high concentration of contaminants is detected, the UAVs tend to fly close to the detection point. The key to detecting the source of contamination is the detection point and nothing else. In Figure 4.7 one of the UAVs moves very close to the source of contamination, but does not find a higher concentration value than the current one.

Finally, we have the behavior of the approach with the Heuristic Dragonfly algorithm. The 4.8 figure shows that, once a high concentration of contaminant was detected, the UAVs are kept close to the detection point, but with more variance. This does not necessarily mean that the UAV doesn't repeat the same destinations. For this reason, statistical analysis is necessary in order to verify the efficiency of the proposed system.

Figures 4.9, 4.10, 4.11 and 4.12 summarize the dispersion of each set of experimental results. Also included are the numerical data that make up the boxplots in tables that are under each figure. In each figure it can be noticed that the median is better, except in the analysis of the proximity to the source of contamination. The best median in this analysis is the Dragonfly Algorithm. Finally, in order to demonstrate that the samples are significant, a single factor ANOVA was carried out. This analysis concludes that the averages of the time, travel and concentration level analyses are significantly different. The proposed strategy is the one with the best statistical values. In the case of proximity to the source of contamination, it is concluded that the averages are the same. In this field, no algorithm did better than the others.

Chapter 5

Conclusions

The present work shows a strategy of search and location of polluting sources of air in a simulated environment in a large area. The strategy is based on the Dragonfly Optimization Algorithm. For the exploration stage, it was proposed to use Hammersley sequences, k-means clustering and a greedy search to find the best exploration route. The use of Hammersley sequences allowed UAVs to cover different points of the search area, avoiding the repetition of sampling points. This work considers a single polluting plume, but the use of Hammersley sequences broadens the probability of finding more sources of contamination.

This work has a coordination approach between the UAVs, also some features differentiate it from similar work. The simulation is done in real time. The search area is 1 km^2. The polluting plume is simulated off-line, loaded into memory at the moment of the UAVs take-off and then mapped in time and space. Table 4.6, from section 4.5, shows the qualitative comparison of the proposed strategy against other similar works.

The main contribution of the work is in the exploitation stage of the Dragonfly algorithm. A Probabilistic component is added to the Separation, Alignment, Cohesion, Attraction to food and Inertia components of the Dragonfly algorithm. The new probabilistic component can be considered as the memory of the algorithm. If an UAV takes a sample of contaminant concentration at one point, the probability of relapse at this point will decrease greatly due to the update of the probabilistic map. The algorithm will prefer, to a certain extent, to go to places with a higher probability of finding a new food value. Another consequence of this is that the UAVs will cover more areas around the position of the food (points with high concentration of pollutants).

The real-time simulation allowed to take in consideration aspects such as UAVs inertia, acceleration and deceleration to move from one point to another. These features will allow an easy implementation in real experiments on the field.

There is enough statistical evidence to say that the results of the proposed approach are better than those of the strategies explained in 3.3. Analyzing the detected concentration level, it was concluded that the means do not differ significantly. So this parameter should not be taken too much into account to measure efficiency. Figure 4.4 shows that cooperative approaches have a better rate of detection of high levels of contaminants. Because of this it can be concluded that the cooperation of each UAV improves the detection rate. This strategy can be used in an environmental or industrial scenario.

5.1 Future work

For the implementation of the strategy in a real platform, or for future work, is recommended:

- **Use a path planning algorithm for traveling through the Hammersley points.** As mentioned in section 4.5, the advantage of Greedy Search is that it performs fewer rotations in Yaw angle before finding a high pollutant measure. If the UAV is able to traverse the Hammersley points without stopping, it will have much more time to explore.
- **XBee modules with high gain antennas could be used for communication.** This ensures communication in the search area between the UAVs.

Bibliography

[1] Alvarado, M., Gonzalez, F., Fletcher, A., and Doshi, A. Towards the development of a low cost airborne sensing system to monitor dust particles after blasting at open-pit mine sites. *Sensors 15*, 8 (2015), 19667–19687.

[2] Benders, S., Wenz, A., and Johansen, T. A. Adaptive path planning for unmanned aircraft using in-flight wind velocity estimation.

[3] Bourne, J. R., Pardyjak, E. R., and Leang, K. K. Coordinated bayesian-based bioinspired plume source term estimation and source seeking for mobile robots. *IEEE Transactions on Robotics* (2019).

[4] Cassano, J., Seefeldt, M. W., Palo, S., Knuth, S. L., Bradley, A. C., Herman, P. D., Kernebone, P. A., and Logan, N. J. Observations of the atmosphere and surface state over terra nova bay, antarctica, using unmanned aerial systems. *Earth System Science Data 8*, 1 (2016).

[5] Colosimo, B., and Senin, N. Geometric tolerances: Impact on product design. *Quality Inspection and Statistical Process Monitoring* (2010).

[6] Di, B., Zhou, R., and Duan, H. Potential field based receding horizon motion planning for centrality-aware multiple uav cooperative surveillance. *Aerospace Science and Technology 46* (2015), 386–397.

[7] Gan, G., Ma, C., and Wu, J. *Data clustering: theory, algorithms, and applications*, vol. 20. Siam, 2007.

[8] Grant, L. L., and Venayagamoorthy, G. K. Swarm intelligence for collective robotic search. In *Design and Control of Intelligent Robotic Systems*. Springer, 2009, pp. 29–47.

[9] Greatwood, C., Richardson, T. S., Freer, J., Thomas, R. M., MacKenzie, A. R., Brownlow, R., Lowry, D., Fisher, R. E., and Nisbet, E. G. Atmospheric sampling on ascension island using multirotor uavs. *Sensors 17*, 6 (2017), 1189.

[10] Greco, F. *Travelling Salesman Problem*. In-teh, 2008.

[11] Greenpeace. El aire que respiro – el estado de la calidad del aire. https://www.greenpeace.org/mexico/publicacion/776/el-aire-que-respiro-el-estado-de-la-calidad-del-aire/, 2018.

[12] Han, J., Xu, Y., Di, L., and Chen, Y. Low-cost multi-uav technologies for contour mapping of nuclear radiation field. *Journal of Intelligent & Robotic Systems 70*, 1-4 (2013), 401–410.

[13] Hosseini, B. *Dispersion of pollutants in the atmosphere: A numerical study*. PhD thesis, Science: Department of Mathematics, 2013.

[14] Jatmiko, W., Sekiyama, K., and Fukuda, T. A pso-based mobile robot for odor source localization in dynamic advection-diffusion with obstacles environment: theory, simulation and measurement. *IEEE Computational Intelligence Magazine 2*, 2 (2007), 37–51.

[15] John, A. A. B., and Dutta, R. Cooperative trajectory planning in an intercommunicating group of uavs for convex plume wrapping. In *Sarnoff Symposium, 2017 IEEE 38th* (2017), IEEE, pp. 1–6.

[16] Khan, A., Yanmaz, E., and Rinner, B. Information merging in multi-uav cooperative search. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (2014), IEEE, pp. 3122–3129.

[17] Koohifar, F., Kumbhar, A., and Guvenc, I. Receding horizon multi-uav cooperative tracking of moving rf source. *IEEE Communications Letters 21*, 6 (2017), 1433–1436.

[18] Kristiansen, R., Oland, E., and Narayanachar, D. Operational concepts in uav formation monitoring of industrial emissions. In *Cognitive Infocommunications (CogInfoCom), 2012 IEEE 3rd International Conference on* (2012), IEEE, pp. 339–344.

[19] Kuipers, L., and Niederreiter, H. *Uniform distribution of sequences*. Courier Corporation, 2012.

[20] Li, S., Guo, Y., and Bingham, B. Multi-robot cooperative control for monitoring and tracking dynamic plumes. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (2014), IEEE, pp. 67–73.

[21] Manjusha, M., and Harikumar, R. Performance analysis of knn classifier and k-means clustering for robust classification of epilepsy from eeg signals. In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (2016), IEEE, pp. 2412–2416.

[22] Mirjalili, S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications 27*, 4 (2016), 1053–1073.

[23] Pang, S., and Farrell, J. A. Chemical plume source localization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 36*, 5 (2006), 1068–1080.

[24] QU, Y., WU, J., XIAO, B., AND YUAN, D. A fault-tolerant cooperative positioning approach for multiple uavs. *IEEE Access 5* (2017), 15630–15640.

[25] ROLDÁN, J. J., LANSAC, B., DEL CERRO, J., AND BARRIENTOS, A. A proposal of multi-uav mission coordination and control architecture. In *Robot 2015: Second Iberian robotics conference* (2016), Springer, pp. 597–608.

[26] ROSSI, M., AND BRUNELLI, D. Autonomous gas detection and mapping with unmanned aerial vehicles. *IEEE Transactions on Instrumentation and Measurement 65*, 4 (2016), 765–775.

[27] SHARMA, V., KUMAR, R., AND KUMAR, R. Quat-dem: Quaternion-dematel based neural model for mutual coordination between uavs. *Information Sciences 418* (2017), 74–90.

[28] SHIMA, T., AND RASMUSSEN, S. *UAV cooperative decision and control: challenges and practical approaches*. SIAM, 2009.

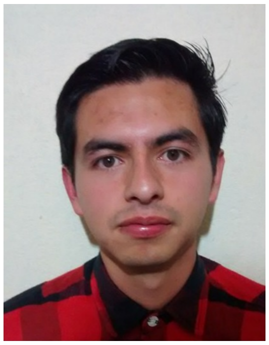
[29] ŠMÍDL, V., AND HOFMAN, R. Tracking of atmospheric release of pollution using unmanned aerial vehicles. *Atmospheric Environment 67* (2013), 425–436.

[30] YUNGAICELA-NAULA, N. M., ZHANG, Y., GARZA-CASTAÑON, L. E., AND MINCHALA, L. I. Uav-based air pollutant source localization using gradient and probabilistic methods. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)* (2018), IEEE, pp. 702–707.

[31] ZHU, S., WANG, D., AND LOW, C. B. Cooperative control of multiple uavs for source seeking. *Journal of Intelligent & Robotic Systems 70*, 1-4 (2013), 293–301.

[32] ZHU, T., HE, W., LING, H., AND ZHANG, Z. Cooperative search strategies of multiple uavs based on clustering using minimum spanning tree. In *International Conference on Sensing and Imaging* (2018), Springer, pp. 112–121.

[33] © 2019 DRONECODE PROJECT, I. Mavlink developer guide. `https://mavlink.io/en/`, 2019.



Andrés Fernando García Calle

Curriculum Vitae

PERSONAL INFORMATION

Birth	September 10 2019
Address	Playa Hornos 3233, Monterrey NL, México
Status	Single
Phone	81 8704 8040
Email	`fernando.garcia88x@gmail.com`

EDUCATION

Bachelor in Electronic and Telecommunications Engineering 2011-2017
Universidad de Cuenca
Cuenca, Ecuador

Master in Science of Engineering 2018-2019
Instituto Tecnológico y de Estudios Superiores de Monterrey
Monterrey NL, México

CERTIFICATIONS

PLC and automation module 2011
Universidad Politécnica Salesiana
Cuenca, Ecuador

Seminar 2016
Escuela Politécnica Nacional

- ”New transmission schemes for the next generation of 5G cellular communications: Mass MIMO and Heterogeneous Networks”.
  Quito, Ecuador.

Certified LabVIEW Associate Developer 2018
ITESM - NI Badge Program

WORK EXPERIENCE

Apprenticeship 2016
Red Sísmica del Austro (RSA)

- Prototype development: charge controller for batteries at RSA monitoring stations;
- Software development: FSK demodulator for charge controller transmissions using an NI-USB and Matlab DAC.

INTEREST AREAS

Embedded systems design, manufacturing and supply chain, engineering, R&D.

PUBLICATIONS

1. Edisson Naula, Andrés F. García, Kenneth Palacio, Luis I. Minchala, Fabian Astudillo-Salinas y Andrés Vazquez-Rodas (2017). Evaluating the Mindwave Headset for Automatic Upper Body Motion Classification, INCISCOS 2017.

SKILLS

Languages	Spanish (Native Language) English (Intermediate)
Software	Matlab, LabVIEW, LaTeX, Python, Bash, Arduino, C, Java, GNS3, Fusion 360, Flutter-Dart.