

IONIAN UNIVERSITY

**Faculty of Information Science
and Informatics**

Department of Informatics



Master's Thesis

Research Directions in Information Technology

**A Study on Area Coverage Methods using Drone
Swarms in Inaccessible Environments**

Konstantinos Bezas

June 18, 2021

Supervisor

Michael Stefanidakis, *Associate Professor,*
Ionian University

Advisory Committee Members

Michael Stefanidakis, *Associate Professor,*
Ionian University

Konstantinos Oikonomou, *Professor,*
Ionian University

Dimitrios Tsoumakos, *Associate Professor,*
Ionian University

This thesis is dedicated to my lovely and supporting parents.

Abstract

Autonomous robotic systems can execute multiple tasks with increased speed and accuracy thus human intervention can be minimized. Advances in technology have led to the increasing adoption of such systems by many industries. Monitoring and information collection from areas of interest are applications that benefit from autonomous robotics. The advanced technology of modern societies paves the way for miniature autonomous areal robotic systems which offer a cost- and time-effective solution to area coverage problems. They are most commonly referred to as drones.

Applications of the last two decades use multiple drones, known as drone swarms, as a solution to tackle problems that either required increased funding or intense human effort. Applications of drone swarms may include, forest fire early detection with drones that fly over the area multiple times during the day and locate possible points of increased temperature. Also, parcel delivery drone swarms and oil spill evolution monitoring are good examples. Human intervention is considered minimal since they only participate in the data analysis when the drones return from the missions.

The main goal of drone swarms is to optimally cover the area of interest with as little human intervention as possible. Sensory devices can provide information relative to the environment and the swarm to navigate the area and successfully avoid obstacles. The drones usually have wireless antennas which help them communicate information with each other to take informed and combined action, and complete a given mission.

Coverage Planning (CPP) is the field which studies the area coverage problem. The basic subcategories of CPP are cellular decomposition and no decomposition. The former separates complex areas with multiple obstacles in smaller and more simple shapes or cells and then creates a path planning for the coverage of each cell. No decomposition is used for simpler areas where the shape is not complex

and obstacles are not present. In the context of the current thesis, an algorithm is developed based on no decomposition for simply-shaped areas. The algorithm also includes the steps which determine the behavior of the swarm when a drone discovers a point of interest and information collection is required.

To evaluate the algorithm’s performance, simulations are developed using the OMNeT++ software. The metrics retained are area coverage time, mean traveled distance per drone for two different covering paths, mean speed, the number of slowdowns required for the collection of information from the points of interest, and the percentage of discovered points. The covering paths studied are, non-overlapping parallel lines –or back-and-forth movement– and spiral.

The results showcase that parallel lines are more effective for the area coverage since the travel time and the distance required to cover the area are lower. This is attributed to the fact that the spiral shape reduces the size of the area’s edges. Each time the coverage of a linear part of the area of interest is finished, the drones remove that part from the area. For the parallel lines coverage, the drones remove similar-sized portions of the area while for the spiral smaller and smaller parts are removed. Due to this behavior, the drones have to switch their formation almost double the times for the spiral shape compared to the parallel lines.

Locating and collecting information from points of interest, based on the algorithm, might require speed reduction. In both parallel and spiral coverage paths, the percentage of located points of interest is similar. The spiral coverage, though, requires an increased number of slowdowns compared to the parallel path. This is attributed to the fact that the drones locate the same points of interest from different perspectives and since they only collect information from the closest point, the effect of increased slowdowns appears.

Περίληψη

Τα αυτόνομα ρομποτικά συστήματα έχουν τη δυνατότητα να εκτελούν διεργασίες με ιδιαίτερα αυξημένες ταχύτητες και ακρίβεια, τέτοιες ώστε η ανθρώπινη παρέμβαση να μην καθίσταται πλέον απαραίτητη. Αυτή η τάση εμφανίζεται πλέον ολοένα και περισσότερο σε όλους τους εργασιακούς τομείς. Σε αυτήν την κατηγορία εντάσσονται και οι εργασίες παρακολούθησης και εκτίμησης της κατάστασης γεωγραφικών εκτάσεων αλλά και η εξερεύνηση περιοχών. Οι τεχνολογικές εξελίξεις των τελευταίων δεκαετιών παρέχουν τη δυνατότητα ανάπτυξης συσκευών μικρού όγκου ικανές να καλύψουν περιοχές από αέρος, τα γνωστά και ως drones.

Σύγχρονες εφαρμογές των drones περιλαμβάνουν την παρακολούθηση δασικών εκτάσεων για πιθανή έξαρση φωτιάς, την παρακολούθηση της εξέλιξης μιας πετρελαιοκηλίδας και την παράδοση παραγγελιών από αέρος. Πολλές από αυτές έχουν την ανάγκη χρήσης πολλαπλών drones τα οποία έπειτα από τη συνεργασία μεταξύ τους αποφέρουν το επιθυμητό αποτέλεσμα. Ο ανθρώπινος παράγοντας σε αυτές τις εφαρμογές περιορίζεται στο σημείο της ανάλυσης των δεδομένων που προκύπτουν έπειτα από την εκτέλεση των αποστολών που ανατίθενται στην ομάδα των drones.

Το βασικό ζήτημα των περισσότερων εφαρμογών είναι η κατά το δυνατόν αποδοτικότερη κάλυψη των περιοχών εντός των οποίων θα κινούνται τα drones με την ελάχιστη δυνατή παρέμβαση του ανθρώπινου παράγοντα. Αυτό σημαίνει πως τα drones θα πρέπει να έχουν τη δυνατότητα να συλλέγουν πληροφορίες από την περιοχή ενδιαφέροντος, να ανταλλάσσουν αυτές που θεωρούν απαραίτητο με τα υπόλοιπα μέλη της ομάδας και να λαμβάνουν τις απαραίτητες αποφάσεις οι οποίες θα οδηγήσουν τελικά στην πλήρη κάλυψη και συλλογή πληροφοριών από την περιοχή ενδιαφέροντος.

Ο τομέας που μελετά το ζήτημα της κάλυψης μιας περιοχής από πολλαπλά ή ένα ρομποτικά συστήματα ονομάζεται Coverage Path Planning (CPP). Αυτός έχει δύο βασικές υποκατηγορίες, κάλυψη σύνθετων σχημάτων με διαχωρισμό της περιοχής σε κελιά και κάλυψη απλούστερων σχημάτων χωρίς διαχωρισμό. Στα πλαίσια της παρο-

ύσας μελέτης, γίνεται ανάπτυξη αλγορίθμου για ομάδα drones με σκοπό την πλήρη κάλυψη περιοχών με απλό σχήμα. Ο αλγόριθμος περιλαμβάνει διαδικασίες που επιτυγχάνουν, με τη χρήση κατανεμημένων ενεργειών και μικρής ποσότητας πληροφορίας, τόσο την πλήρη κάλυψη περιοχών όσο και τη συλλογή πληροφοριών από σημεία ενδιαφέροντος που εντοπίζονται σε πραγματικό χρόνο. Ο αλγόριθμος περιλαμβάνει επίσης, διαδικασίες προσαρμογής της ταχύτητας της ομάδας με σκοπό την αποδοτικότερη συλλογή δεδομένων από τα σημεία ενδιαφέροντος που εντοπίζονται.

Προκειμένου να γίνει εκτίμηση της απόδοσης του αλγορίθμου αναπτύχθηκαν προσομοιώσεις όπου πραγματοποιήθηκαν μετρήσεις σχετικά με το χρόνο κάλυψης της περιοχής, τη μέση ταχύτητα, τον αριθμό πτώσεων ταχύτητας που πραγματοποιήθηκε κατά την κάλυψη, το ποσοστό των σημείων ενδιαφέροντος από όπου έγινε πλήρης συλλογή δεδομένων αλλά και την απαιτούμενη απόσταση που διήνυσαν τα drones με βάση δύο διαφορετικά σχήματα κάλυψης. Τα σχήματα κάλυψης που μελετήθηκαν είναι η σπείρα και η κάλυψη σε παράλληλες και μη επικαλυπτόμενες γραμμές.

Βάσει των αποτελεσμάτων, φαίνεται πως η κάλυψη της περιοχής με τη χρήση των παράλληλων γραμμών απαιτεί λιγότερο χρόνο πτήσης όπως και μικρότερη απόσταση για την ολοκλήρωση της κάλυψης συγκριτικά με τη σπείρα. Αυτό συμβαίνει καθώς με τη σπείρα κάθε φορά που καλύπτεται μία πλευρά της περιοχής, αυτή αφαιρείται από το σχήμα με αποτέλεσμα να μικραίνει κάθε φορά μία πλευρά του σχήματος. Έτσι, η ομάδα των drones χρειάζεται να αλλάζει σχηματισμό περισσότερες φορές συγκριτικά με τις παράλληλες γραμμές κάλυψης κάτι που κοστίζει σε χρόνο και απόσταση.

Ο εντοπισμός του ίδιου αριθμού σημείων ενδιαφέροντος, για τη σπειροειδή κάλυψη της περιοχής, απαιτεί τη μείωση της ταχύτητας περισσότερες φορές συγκριτικά με τις παράλληλες γραμμές. Αυτό αποδίδεται κυρίως στην κατανομή των σημείων όπως και στην κατεύθυνση των drones η οποία στη σπειροειδή κάλυψη για κάθε αλλαγή σχηματισμού αλλάζει κατά 90° ενώ για τις παράλληλες γραμμές αλλάζει κατά 180° . Και για τα δύο σχήματα ισχύει ότι η ομάδα των drones ήταν ικανή να εντοπίσει μεγαλύτερο αριθμό σημείων ενδιαφέροντος για μικρότερες τιμές της ακτίνας κάλυψης. Αυτό αποδίδεται στη λειτουργία του αλγορίθμου η οποία ορίζει πως ένα drone συλλέγει πληροφορίες από ένα σημείο κάθε φορά και συγκεκριμένα από αυτό που βρίσκεται πλησιέστερα στο drone.

Ευχαριστίες

Πρωτίστως, θα ήθελα να ευχαριστήσω τους γονείς μου για την υπομονή και την αγάπη τους που μου προσέφεραν σε όλη μου τη ζωή. Αυτοί είναι ο λόγος για τον οποίο βρίσκομαι εδώ σήμερα και τους οφείλω τα πάντα για τη στήριξή τους. Μοναδική τους έννοια ήταν πάντοτε η πρόοδος των παιδιών τους. Ευχαριστώ και τον αδερφό μου Γιάννη για τη στήριξη και την αγάπη του.

Θα ήθελα να ευχαριστήσω θερμά τον κ. Στεφανιδάκη για την καθοδήγηση και τη βοήθειά του προς την επιτυχή ολοκλήρωση της παρούσας διπλωματικής εργασίας. Επίσης, θα ήθελα να ευχαριστήσω τα μέλη της τριμελούς επιτροπής για τη βοήθεια και τις οδηγίες που προσέφεραν σχετικά με τις βελτιώσεις της διπλωματικής εργασίας.

Ιδιαίτερες ευχαριστίες θα ήθελα να απευθύνω στον κ. Κωνσταντίνο Οικονόμου ο οποίος επί τρία συναπτά έτη, έκανε ό,τι ήταν ανθρωπίνως δυνατό για την προσωπική μου πρόοδο και εξέλιξη. Δίπλα του απέκτησα ομαδικό πνεύμα, αίσθημα ευθύνης και ένα πολύ μεγάλο μέρος των γνώσεών μου. Του είμαι εξαιρετικά ευγνώμων για όλη τη βοήθεια και την καθοδήγηση που μου προσέφερε, για την υπομονή και την επιμονή του να μου διδάξει ό,τι περισσότερο μπορούσε τόσο σε ακαδημαϊκό επίπεδο όσο και σε ζητήματα συμπεριφοράς και ήθους προς τους συναδέλφους και προς την ομάδα.

Δε θα μπορούσε να μη γίνει αναφορά στο Γεώργιο Τσουμάνη ο οποίος από το προπτυχιακό ακόμη προσέφερε πολλές φορές βοήθεια και καθοδήγηση για ζητήματα ερευνητικά. Η βοήθειά του δεν έλειψε και από την παρούσα διπλωματική. Τον ευχαριστώ θερμά.

Τέλος, επειδή χωρίς καλή παρέα και καλούς συνεργάτες ο δρόμος είναι παντοτε πιο τραχύς, οφείλω να ευχαριστήσω τους συναδέλφους και πολυ καλους φίλους Αλέξανδρο, Αστέριο, Κορίνα, Σπύρο, Ανδρεάνα και Γιάννη για όλη την τρέλα που ζήσαμε μαζί τα τελευταία τρία χρόνια. Ήταν ένα μοναδικό ταξίδι από εμπειρίες, χαμόγελα, τσακωμούς συνεργατικές εργασίες και ακαδημαϊκές δραστηριότητες. Υπήρξαμε ομάδα από τις πρώτες στιγμές της πτυχιακής μας εργασίας και έκτοτε πορευόμαστε έτσι. Τους

vi

ευχαριστώ πολύ!

Contents

A	Introduction	1
A.1	Problem Definition	3
A.2	Related Work	4
A.2.1	Base Stations and Swarm Autonomy	5
A.2.2	Path Scheduling on Areas with Known Points	6
A.2.3	Area Coverage on Unknown Environments	7
A.3	Contribution	8
A.4	Thesis Outline	8
B	Coverage Path Planning	10
B.1	Area of Interest and Decomposition Techniques	10
B.2	Area Coverage with no Decomposition	12
C	A Drone Self-Organized Swarm Algorithm	14
C.1	Drones Swarm Self Organization	15
C.1.1	Aligning with Borders	16
C.1.2	Partial Linear Coverage	19
C.2	Collecting Information from Points of Interest	20
C.2.1	Distance Between Drone and Point of Interest	21
C.2.2	Distance within Radius	21
C.2.3	Speed Adjustment	23
D	Simulation Results	24
D.1	Simulations Setup	24
D.2	Performance Evaluation	28

D.2.1	Area Coverage Path	28
D.2.2	Variable Scanning Radius Values	32
D.2.3	Data Collected	41
D.3	Results Summary	46
E	Conclusions and Future Work	49
E.1	Conclusions	51
E.2	Future Work	51
	Bibliography	52
I	Notations	61
II	Abbreviations	62
		62
III	Source Code	63
	Index	68

List of Figures

B.1	Area of interest using exact cell decomposition in order to divide the area into smaller and more simple shapes or cells while avoiding the no-fly zones: (a) Trapezoidal cell decomposition. (b) Boustrophedon cell decomposition.	11
B.2	Area coverage path planning in simple polygons where no decomposition techniques are applied: (a) Parallel lines coverage path planning. (b) Spiral coverage path planning.	13
C.1	Locating a PoI and calculating the distance in scanning radius d_s . . .	22
D.1	A drone covering an area and discovering a PoI at the most remote point of its scanning radius.	26
D.2	The simulated environment in OMNeT++ where 1 drone with 50 meters of scanning radius is depicted. The area of interest is within the white borders where points of interest are also depicted.	27
D.3	The simulated environment in OMNeT++ where 2 drone with 50 meters of scanning radius is depicted. The area of interest is within the white borders where points of interest are also depicted.	27
D.4	The simulated environment in OMNeT++ where 4 drone with 50 meters of scanning radius is depicted. The area of interest is within the white borders where points of interest are also depicted.	28
D.5	Examples of parallel lines and spiral area coverage. The yellow line depicts the path of the drone: (a) Parallel lines area coverage. (b) Spiral area coverage.	29

- D.6 The mean distance a drone traveled during the simulations for a swarm size equal to 1, 2 and 4 drones based on the scanning radius r_s : **(a)** Depicts the mean distance per drone while the r_s changes in size. **(b)** The number of switched formations for parallel and spiral coverage paths based on variable r_s 31
- D.7 Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 25m$ and parallel lines coverage path: **(a)** Time required for the complete area coverage by the initial number of PoIs. **(b)** The number of slowdowns required for the information collection from the PoIs. Depends on the number of PoIs that exist inside the map. **(c)** Mean speed throughout the simulation with variable number of initial PoIs. Depends on the number of slowdowns. **(d)** The percentage of scanned PoIs while they increase. 33
- D.8 Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 25m$ and spiral coverage path: **(a)** Time required for the complete area coverage by the initial number of PoIs. **(b)** The number of slowdowns required for the information collection from the PoIs. Depends on the number of PoIs that exist inside the map. **(c)** Mean speed throughout the simulation with variable number of initial PoIs. Depends on the number of slowdowns. **(d)** The percentage of scanned PoIs while they increase. 34
- D.9 Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 50m$ and parallel lines coverage path: **(a)** Time required for the complete area coverage by the initial number of PoIs. **(b)** The number of slowdowns required for the information collection from the PoIs. Depends on the number of PoIs that exist inside the map. **(c)** Mean speed throughout the simulations with variable number of initial PoIs. Depends on the number of slowdowns. **(d)** The percentage of scanned PoIs while they increase. 35

- D.10 Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 50m$ and spiral coverage path: **(a)** Time required for the complete area coverage by the initial number of PoIs. **(b)** The number of slowdowns required for the information collection from the PoIs. Depends on the number of PoIs that exist inside the map. **(c)** Mean speed throughout the simulations with variable number of initial PoIs. Depends on the number of slowdowns. **(d)** The percentage of scanned PoIs while they increase. 36
- D.11 Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 100m$ and parallel lines coverage path: **(a)** Time required for the complete area coverage by the initial number of PoIs. **(b)** The number of slowdowns required for the information collection from the PoIs. Depends on the number of PoIs that exist inside the map. **(c)** Mean speed throughout the simulations with variable number of initial PoIs. Depends on the number of slowdowns. **(d)** The percentage of scanned PoIs while they increase. 37
- D.12 Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 100m$ and spiral coverage path: **(a)** Time required for the complete area coverage by the initial number of PoIs. **(b)** The number of slowdowns required for the information collection from the PoIs. Depends on the number of PoIs that exist inside the map. **(c)** Mean speed throughout the simulations with variable number of initial PoIs. Depends on the number of slowdowns. **(d)** The percentage of scanned PoIs while they increase. 38
- D.13 Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 25m$ and parallel lines coverage path. The figures depict metric relative to the data collected by the drones during the mission. The data is measured in MegaBytes (MB): **(a)** Mean data per drone for missions with variable number of PoIs. **(b)** Total data collected by the swarm. 42
- D.14 Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 25m$ and spiral coverage path. The figures depict metric relative to the data collected by the drones during the mission. The data is measured in MegaBytes (MB): **(a)** Mean data per drone for missions with variable number of PoIs. **(b)** Total data collected by the swarm. 44

- D.15 Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 50m$ and parallel lines coverage path. The figures depict metric relative to the data collected by the drones during the mission. The data is measured in MegaBytes (MB): **(a)** Mean data per drone for missions with variable number of PoIs. **(b)** Total data collected by the swarm. 44
- D.16 Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 50$ and spiral coverage path. The figures depict metric relative to the data collected by the drones during the mission. The data is measured in MegaBytes (MB): **(a)** Mean data per drone for missions with variable number of PoIs. **(b)** Total data collected by the swarm. 45
- D.17 Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 100m$ and parallel lines coverage path. The figures depict metric relative to the data collected by the drones during the mission. The data is measured in MegaBytes (MB): **(a)** Mean data per drone for missions with variable number of PoIs. **(b)** Total data collected by the swarm. 45
- D.18 Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 100m$ and spiral coverage path. The figures depict metric relative to the data collected by the drones during the mission. The data is measured in MegaBytes (MB): **(a)** Mean data per drone for missions with variable number of PoIs. **(b)** Total data collected by the swarm. 46

List of Algorithms

1	Drone Swarm Border Align for Scanning	18
2	Partial Area Coverage in Linear Movement	19

Chapter A

Introduction

DRONES nowadays are used for variable applications. The technological advances, mainly in sensory devices and nanotechnology, provide the tools that enable the development of miniature drones which can last for several minutes to hours per flight or even fly continuously. Historically, drones were funded by the US military with the first flights dating back to 1903 [1]. During the first demonstrations, drones had very limited features. They were remotely operated by one person and wireless communications were used to transmit basic commands. The most attractive aspect of this technology was the fact that it was unmanned which meant that no pilot was risking his life on the battlefield. By the year 1940, in WWI, all major powers started using drones.

Nowadays, drones have advanced to the point where no human operator is required to complete a task. They are being adopted by many industries since they can potentially lower the cost of product delivery and minimize human casualties under states of emergency. Drone technology can be applied to many aspects of human life. A few examples are, post-earthquake response for human detection and damage assessment [2], early detection of forest fires [3], wildfire tracking [4], parcel or food delivery systems [5, 6], structural integrity inspection [7] and powerline inspection [8]. Major companies such as Amazon have shown interest in swarms and aim to release actual parcel delivery systems.

Ever since the miniaturization of computer components, sensory devices have become much easier to include in a compact and easy-to-develop system with a basic operation such as the collection of information from the surrounding environment

and wireless data transmission. Such systems can be adjusted to operate on drones and even create swarms that can be coordinated to complete tasks as a team. The traditional operation of a drone requires one pilot. This model cannot be employed for drone applications since a typical swarm can include tens of drones, hence the system is prone to failure due to human error.

The swarm collects information in real-time and decides the necessary actions towards the completion of a mission. A major issue with drone swarms is the navigation in a natural environment. Algorithmic steps can provide a path planning scheme which can be altered in real-time. The information about which path a drone has to follow is not hard-coded, rather an algorithm, which includes simple steps individually followed by the drones, provides the necessary distributed behavior. The algorithm has to include accurate and well-defined processes to avoid any physical collisions between the drones of the swarm.

The role of an operator in drone swarms can be either absent or in the form of simple commands [9] which coordinate the entire swarm towards a task completion. During the last decade, applications with multiple drones are headed towards full autonomy. This can provide less overhead since the drones are coordinated using a behavior obtained by the algorithm, while the operator is mainly focused on accomplishing a given task rather than having to control multiple drones to navigate in an area.

Precision agriculture applications also benefit from the use of drone swarms [10]. A large amount of pesticides and insect-killing chemicals is used to provide a higher yield. Improper use and manual application of such materials have led to diseases in about 1 million people based on information published by the World Health Organization. Harmful chemicals and pesticides do not pose any threat to drones. Thus, their application can be done using drone swarms.

Applications in open seas such as Search and Rescue (SaR) [11], and oil-spill modelling [12] or real-time mapping [13] can also benefit from swarm algorithms and drone swarms in general. SaR missions and oil-spill modeling or mapping are time-critical applications. In the case of SaR missions, multiple scenarios can be assumed, such as emergency plane landing at seas or boat sinking events. Both cases require the full of large open-seas areas a task well suited for a swarm of drones given the fact that sensors such as thermal cameras are very commonly used tools.

drone swarms can be applied to scenarios such as forest fire-fighting [14, 15] or early forest fire detection [16]. Both applications require a well-defined algorithm to cover the area efficiently and provide low response times. The only initial input to the swarm is the geographical . The swarm can operate with minimal to no intervention by an operator. The drones can be launched by a , reach the area of interest and once the coverage is complete, they can return and offload all the information they collected to the base station. Geographical information can be provided to the swarm by satellite.

A.1 Problem Definition

Swarm autonomy is a major issue excessively studied throughout the literature [17, 16]. More specifically, pathfinding and algorithms are crucial for the proper information collection from an unknown area. The main hurdle the swarm has to face is which path to choose to completely cover an area for which most likely no information is available.

In the case where there are specific Points of Interest (PoIs) within the area and the main goal is the collection of information for these PoIs, the route each drone has to follow can be calculated before the mission begins and the drones will simply follow a specific path they have been coordinated at. The main issue with such an approach is that real-world scenarios include much more abstract environments where very limited to no information is available. In such cases, the swarm requires more flexible algorithms which can provide an adaptive behavior to meet the requirements of an unknown area.

The algorithm has to provide well-defined steps towards behavior that helps the drones avoid collisions between them while at the same time fully cover the area that they are required to. Many factors need to be taken into account. For instance, the movement of the swarm is required to be as efficient as possible in terms of travel distance. Variable area shapes require different flying routes (or coverage paths) to fully and efficiently cover the area. Different coverage paths require precise formation switches which ensure similar travel distance for all drones.

Equally as important is covering the area exactly once. This is achieved by creating routes for each drone that do not overlap. The discovery of PoIs and the information collection from them is usually the main goal for most applications.

These processes vary based on the effective area or scanning radius of each drone. Changes in altitude can affect the scanning radius.

The main focus of the current study is the coverage of unknown areas and the detection and information collection from PoIs within the area. In the following section, past related work is provided.

A.2 Related Work

Many studies over the years focus on drone swarms autonomy [18] and coordination using base stations [19]. The typical model of operation and coordination of drones remotely by a human pilot is slowly becoming part of the past. The move towards autonomy is the natural next step since digital systems can provide higher levels of accuracy since drones can be equipped with sensors [20] which provide precision in natural environments. Also, drones can be tethered to nearby smart devices [21] which can provide location information effectively aiding towards the proper localization of drone swarms. This localization scheme can help a swarm conserve energy resources since the localization does not require communication with satellites rather small-range, low-energy transmissions which can reach nearby devices.

detection during the area coverage is an important goal of many studies which employ various technologies such as image-based deep learning and computer vision techniques [22, 23]. Nature-inspired algorithms have shown to provide very accurate results. A study published in 2018 [24] suggests an algorithm that can be adopted by a swarm of drones that use various environment-sensing techniques such as cameras or wireless signal detection. The study suggests the use of three nature-inspired algorithms, stigmergy, flocking, and evolution. Stigmergy is a socializing mechanism based on pheromones used by insects and the algorithm uses this logic in order to attract drones in specific points of interest.

Many applications include area monitoring using WSN-based systems [25, 26, 21]. Wireless sensory devices can be deployed on a wide area and relay sensed information back to a base station. While this solution can provide accurate results, maintenance cost and possible failure can be a deterring factor. Typical WSNs are comprised of battery-powered nodes since the construction of a wired infrastructure requires large investments [27].

Wireless Sensor Networks are much easier to deploy. Studies have shown that

by combining multiple algorithms such as the Particle Swarm Optimization (PSO) and Voronoi diagram [28] efficient WSN deployment can be achieved. Yet, WSNs operate using batteries and wireless networking algorithms which can be severely affected when a node has to communicate large amounts of information to help the network's services remain active. This is known as the energy hole problem [29]. Drone swarms can potentially eliminate the issues that arise from statically deployed WSNs since they provide mobility, hence they are prone to system alterations—such as the inclusion of additional sensors or the maintenance of sensitive components—much easier.

A.2.1 Base Stations and Swarm Autonomy

Various ()-based systems exist, all ranging in terms of communication capabilities and sensory devices included on the drones. This is attributed to the fact that different applications have unique requirements depending on which piece of environmental information is considered important. For instance, early fire detection and monitoring of wooded areas using drone swarms require cameras equipped with infrared and visible light sensors [16, 30], while a model for area monitoring using drone swarms as mobile sink nodes which collect information from an installed WSN [31] might have requirements for different wireless antennas.

Base stations are computers that can serve as the operator/coordinator for the drone swarms, effectively removing the requirement of human operators [32]. A base station needs to be able to adapt to various swarm-based systems since they are required to coordinate swarms which usually range in terms of capabilities. Base stations can, also, provide an interface between the different swarms. Building multiple base stations could be part of the solution to controlling multiple swarms but the cost is considered to rise, hence a single base station that can operate various swarms is considered a more cost-effective solution.

Drone swarm architectures can also include drones with various characteristics within the swarm [33] that can be used for crisis response applications such as major earthquake events. The architecture can include drones that operate at different altitudes and can serve a unique purpose. For instance, lower altitude Vertical Take-Off and Landing (VTOL) drones can collect information on sites where buildings have collapsed by employing deep penetrating radars or infrared cameras to locate

potentially trapped people. Drones of higher altitudes can provide real-time mapping of the area to assess the damage that has been inflicted on the area. And blimps –high altitude balloons– due to their size can serve as the communications backbone for real-time collection of data from the swarm.

Decentralized algorithms can provide the necessary steps towards perimeter surveillance using a swarm of drones [34]. The drones are required to operate with information relative to nearby members of the swarm. No coordination is provided from outside the swarm. The drones have a single goal, monitor the entire perimeter using simple behavior by communicating with nearby drones. The drones occasionally require refueling since their flight time is limited, hence the algorithm provides dynamic behavior for drone removal and re-entrance in the area. When the surveillance begins, all drones fly towards the same direction in a line defined by the borders. Each time a drone meets another, they both change direction. This scheme continues until all drones obtain one portion of the borderline. This portion is prone to change due to the addition or removal of drones during surveillance missions.

A.2.2 Path Scheduling on Areas with Known Points

Applications that include area coverage where points of interest are known before the mission begins, can be categorized as Travelling Salesman Problem (TSP) for single path finding or Vehicular Routing Problem (VRP) where multiple routes are required [35, 36]. Generally, combined behavior from multiple drones or a swarm of drones is referred to as Swarm Intelligence. This can include the path scheduling for all drones as well as drone response to unexpected events such as handling an obstacle and path rescheduling to avoid it.

Problems such as the TSP and VRP are NP-Complete optimization problems. Algorithms for team-based goal completion can also be found in nature. Bee colonies use simple processes to achieve their main goal which is honey production [37]. Bee Colony Optimization (BCO) has shown great potential towards solving optimization problems [38, 39, 21]. Agents which represent bees, follow simple steps and interactions which provide small portions of the required solution and thus contribute to solving larger-scale problems.

Ant intelligence [40, 41] has been observed to provide successful results in ant colonies. More specifically, ants can cooperate very efficiently without requiring

complex behavior and work as a team to build very intricate constructions where they store their food. The ant intelligence algorithm has shown potential for problems such as TSP [42] and VRP [43]. By providing approximate solutions to such problems, drones swarm applications where area coverage with known points of interest is required can greatly benefit [44].

A.2.3 Area Coverage on Unknown Environments

Area coverage in environments with very little knowledge cannot be achieved with the methods described above. Rather, approaches that define algorithmic processes that aim towards the full area coverage and collection of information are required [45]. This problem is also known as the () problem and it can be separated into three categories based on whether the area of interest is split into subareas, which are covered separately, or a grid is used and when a drone reached the center of a square this is considered covered. The third category is called no decomposition where drones follow a path which covers the entire area without separating it. In the first case, the method is referred to as approximate cellular decomposition [46] while in the second, exact cellular decomposition [47].

Drones with miniature sizes can form a swarm which is ideal for indoor environment exploration. A study published in 2019 by McGuire *et al.* [48] describes an algorithm that can be employed for indoor exploration. It is based on area coverage concepts. The drones that form the swarm fly in multiple directions to cover the area more efficiently. Their starting point is a small base station which is used to attract the drones back when required. This is achieved using radio transmissions. When the drones receive the signal, they locate the point it originated from and fly towards that direction. The navigation in the area is achieved by tracking and following the walls.

The coverage path followed by the drones, is an important aspect of area coverage and information collection. Studies have shown the effects of more rounded shapes such as the spiral and shapes with multiple edges [49]. This 2020 study has proven that coverage shape is closely related to the shape of the examined area. Areas with more edges, which tend to the circle, are being covered more efficiently from rounded coverage shapes, while shapes with fewer edges that tend to the square are better covered from parallel lines. Applications that require data collection from

WSN have increased efficiency by using circular coverage shapes [50].

A.3 Contribution

The current study focuses on the full coverage of unknown areas and the collection of information from discovered PoIs in real-time. An algorithm is developed which coordinates a swarm of drones towards the full area coverage. The algorithm is based on CPP with no decomposition for simple areas with no obstacles included and the main goal is to cover the entire area and collect as much information as possible. First and foremost, the drones need to have the ability to navigate a new area with the minimum information required. This information is provided during the initialization process and it describes the area's borders. Information relative to the swarm is obtained in real-time using beacon transmissions which inform the entire swarm about each drone's current state.

The routes are also created in real-time and they are based on the number of drones currently active in the swarm. This detection takes place the moment all drones start operating. The swarm creates a linear formation and all drones maintain a stable distance from each other. A circular scanning radius is assumed which is used for proper distance maintenance. The coverage of the area is executed in linear movements and each drone covers an area equal to the traveled distance multiplied by the scanning diameter.

The algorithm also includes processes that state the way a drone collects information from PoIs. Each time a PoI enters the scanning radius, while a drone is moving, the drone calculates the amount of information it requires from that PoI, and based on the amount of time the PoI will remain within the scanning radius, it adjusts its speed accordingly. The speed is transmitted to the rest of the swarm which also adjusts its speed. This helps the drones remain in proximity.

A.4 Thesis Outline

The rest of the thesis is organized as follows. Chapter B describes the field relative to the CPP problem. Decomposition techniques, exact and approximate are analyzed. Also, CPP with no decomposition is described. The latter is employed by the

current algorithm.

Chapter C includes the proposed algorithm, as a solution to the unknown area coverage problem, is detailed. The algorithm is separated into two parts, the area coverage and the PoIs detection and information collection. The area coverage part can be separated into two phases. During the first (Subsection C.1.1), drones in the swarm locate each other and exchange information relative to their current location. This location remains unchanged until all drones decide which initial and final points to claim. The next phase is described in Subsection C.1.2. The drones start covering the area while maintaining connectivity to the rest of the swarm to adjust their behavior accordingly. Next, Subsection C.2.2 details about the distance of a PoI in the radius are provided. The chapter concludes with Subsection C.2.3 which includes the speed adjustment steps that help a drone fully obtain all the required information from a located PoI.

Chapter D describes the simulated environment as well as the simulations and the results that provide an insight into the algorithm's performance. In more detail, Section D.1 includes all the information relative to the simulation software and the setup of the environment. It describes the variables that are used to calibrate it as well as a short description of the algorithm's edge cases discovered during the simulations. Section D.2 includes figures which depict the graphical user interface of the simulated environment. In the first Subsection D.2.1 a detailed analysis of the swarm's travel distance is described. Next, Subsection D.2.2 provides results and observations relative to the time required for the area coverage as well as the swarm's mean speed, the number of slowdowns required for the information collection, and the percentage of PoIs successfully collected. Finally, Subsection D.3 summarizes the results and provides conclusions and observations relative to the algorithm's performance.

Finally, conclusions are derived and possible future work is presented in Chapter E.

Chapter B

Coverage Path Planning

COVERAGE Path Planning (CPP) [45] is the problem definition where a single drone or a swarm of drones attempt to fully an area to collect important information or map it. The area might also include subregions where the drones are not allowed to fly over. This could be true for areas where tall buildings exist or aircraft landing zones make drone flights a dangerous endeavor. The area is split into non-intersecting regions called , the size of which varies based on the drone's area coverage or . This is also called cell resolution. The drone's coverage defines the size of the cell. Smaller cells can potentially be covered in a single move while larger ones that exceed the drone's scanning radius, might require multiple moves to cover it. A strategy to ensure full area coverage is required.

B.1 Area of Interest and Decomposition Techniques

The area of interest is defined by a polygon that has at least three vertices and the edges which connect them. Typically, square or parallelogram shapes are used to define the area of interest. Polygons with more vertices and edges can also be used. The angles created at the places where the edges are connected can be used for calculations relative to the path followed by the drones. Variable angle sizes can create polygons of different shapes each of which has specific requirements in terms of . No-fly regions need to be taken into account and specific behavior is required to ensure full area coverage while bypassing these areas. Complex polygons can be

divided into smaller regions defined by simpler shapes that are easier to cover.

This technique is called . It ensures easier area coverage since the drones have to cover smaller portions of the original area which have simpler shapes. Exact cellular decomposition is a subcategory of the decomposition technique which is based on back-and-forth –or lines– path which focuses on finding the next cell to be covered [51]. There are two major exact decomposition categories, trapezoidal and boustrophedon. The categorization refers to the shape of cell decomposition. The trapezoidal takes into consideration the no-fly zones’ vertices effectively decomposing the area of interest into smaller trapezoidal shapes. In the case of boustrophedon decomposition, the area of interest is split using vertical lines which create a cell where the no-fly zone is included entirely and two areas on both sides of the initial cell. Figure B.1 depicts the area decomposition for both trapezoidal as well as boustrophedon techniques.

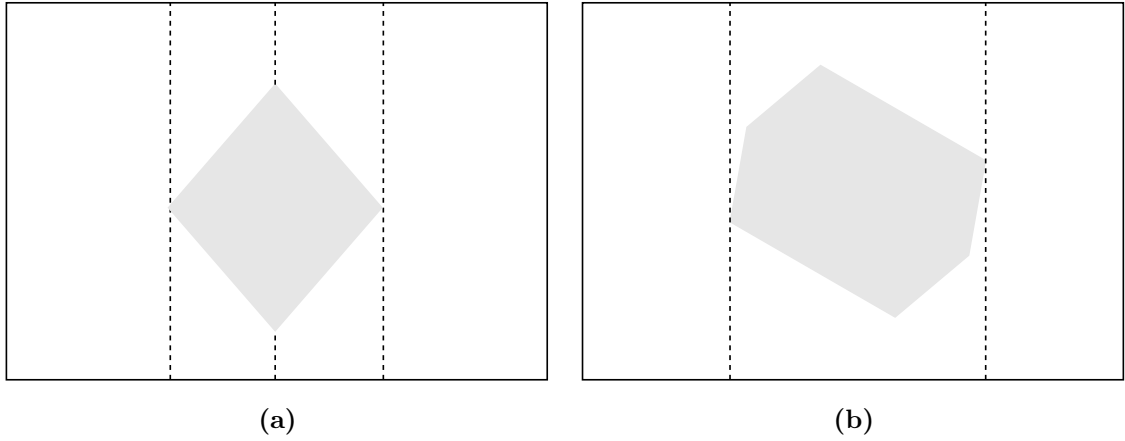


Figure B.1: Area of interest using exact cell decomposition in order to divide the area into smaller and more simple shapes or cells while avoiding the no-fly zones: **(a)** Trapezoidal cell decomposition. **(b)** Boustrophedon cell decomposition.

The approximate cellular decomposition uses a different approach to solve the CPP problem. A grid is created in the area of interest which does not account for any possible obstacles. Rather, the drones have to avoid grid cells that overlap even by a small portion with the no-fly zone. A grid cell size is defined based on the drone’s ability to sense the area. For example, if a drone uses a camera and flies at a specified altitude, the cell size equals the camera’s field of view or in the case of

wireless transmissions where the drone has to locate wireless transmitting devices such as a WSN's nodes, the cell size equals the transmission radius of the wireless antennas. Coverage paths can be obtained using grid-based algorithms. A grid cell is considered covered when the drone reaches its center.

These techniques can potentially provide a solution to the CPP problem but the choice depends on the amount of available information about the area of interest. Environments with relative stability such as a city can be covered with one of the two cell decomposition techniques. An application that requires continuous surveillance of cities can take advantage of the city design, hence it can have information about the no-fly zones and adjust the path planning accordingly. Such applications using available information to create a coverage path have three phases, decomposition, planning, and execution. The coverage path is calculated during the planning phase and is affected by the required performance metric (i.e. the predefined metric based on which the path has to be adjusted to, for example, shortest or mission time minimization).

B.2 Area Coverage with no Decomposition

An area of interest with simple shapes such as parallelograms can have a path planning scheme that does not require complex calculations or any kind of decomposition. This is true especially in cases where the area has no obstacles to intervene with the calculated path. Simple parallelogram areas can be covered using paths that ensure full coverage and based on the area the drones can cover at a time, can ensure the shortest path by which full coverage is achieved. Figure B.2 depicts two coverage paths on a simple polygon area. Figure B.2a depicts the back-and-forth or parallel lines coverage path and Figure B.2b depicts the coverage path.

Coverage shapes of this type are especially important in applications where points of interest need to be discovered within an area. The drones follow a simple path that may contain multiple changes or maneuvers. Usually, the final coverage path consists of multiple straight-line paths and maneuvers which create the final path covering the entire area of interest. In the context of the current thesis, the parallel lines and the spiral shapes are examined in simple polygon areas with techniques applied. Also, the discovery of points of interest is examined and results are derived on the algorithm's efficiency. In the next chapter, an algorithm for a swarm self-

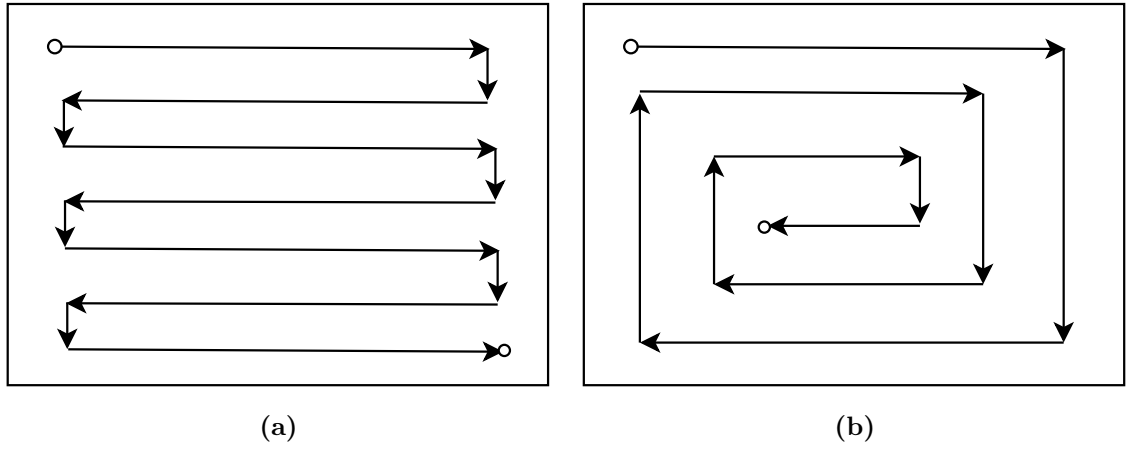


Figure B.2: Area coverage path planning in simple polygons where no decomposition techniques are applied: **(a)** Parallel lines coverage path planning. **(b)** Spiral coverage path planning.

organization is proposed.

Chapter C

A Drone Self-Organized Swarm Algorithm

THE current chapter details the algorithmic process developed in the context of the current thesis to tackle the area coverage problem. The algorithm includes processes for distributed and self-organizing tasks for applications where single or multiple drones are required. The development of the algorithm is based on the principle that minimal to no should be required for the proper operation of the drones. The latter should be able to self-organize using wireless transmissions during monitoring missions as well as distributed operations that employ information relative to the swarm, for instance, relative and of .

A key feature built into the algorithmic logic is that the drones are mainly or devices which in turn means that the currently monitored area is, for the most part, an unknown or inaccessible to humans territory. Consequently, the drones are required to operate with tiny amounts of information provided as initial input. This information describes the relative area for which monitoring or scouting actions need to be taken. Further information, that aid in the proper swarm operation, is acquired in real-time during missions.

The main goal of the algorithm is to provide the drones with the necessary steps which lead to the complete area coverage and the detection of possible PoIs as well as the collection of information describing the state of each one. These steps have been carefully developed to make good use of small amounts of local and swarm information that can lead to complete area coverage without requiring any compute-

or memory-intensive operations. As a result, each drone participating in the swarm requires central processing units with minimal capabilities that can meet the small algorithmic demands.

Based on the algorithm's main functionality, it is derived that a swarm of drones does not require any central management either from a human or a base station to scan an unknown geographical area where no PoIs are known before the mission. The swarm can cover the entire area and collect relevant information from any PoI the drones consider important. Also, the path each drone follows is based purely on information collected in real-time from the entire swarm and consequently does not require pre-calculation and can be altered on demand. The algorithm can operate at square and rectangle polygons, where the internal angle of any two sides does not exceed 90° .

The following sections provide a deeper insight into the main operations of the algorithm. Section C.1 details the algorithmic steps based on which every drone in the swarm can take an informed decision about its path without intercepting other drones. This Section is separated into two (2) Subsections C.1.1, C.1.2 where the formation and the full area coverage are described respectively. Section C.2 contains the analytical steps for the information collection process. The process is split into three (3) separate parts. In Subsection C.2.1 the distance calculation between the drone and a PoI is analyzed. Subsection C.2.2 includes details relative to the calculation of the amount of time and distance a PoI remains within the scanning radius. Finally, Subsection C.2.3 details the speed adjustment steps for the swarm.

C.1 Drones Swarm Self Organization

The basis upon which the algorithm can provide the necessary steps towards full area coverage is the of the entire swarm. Controlling a swarm comprised of multiple drones is a very intricate task. In the case of human operators, their number would have to match the number of drones since one person can only operate a single drone at a given moment. Whereas, in the case of base station operation, a very capable computer would have to remotely operate a swarm of drones. In both cases, the swarm requires continuous connectivity with the base station at all times since decisions are taken centrally and the drones simply execute the coordinated actions.

These factors introduce additional complexity to the entire system.

Self-organization can potentially solve most of the issues remote operation introduces but can be an intricate development task. It requires operations that enable each drone in the swarm to accomplish simple individual tasks while maintaining connectivity with the rest of the swarm. The exchange of information between drones ensures the seamless operation of the entire swarm.

C.1.1 Aligning with Borders

The basis of proper operation of the swarm is the self-organization steps. The first step towards achieving the monitoring of an area is providing safe steps for the proper alignment of the drones. The alignment in the current algorithm is achieved using the area borders to provide full coverage given a predefined scanning radius. Lets consider a swarm with n drones. Each drone is provided with an initial amount of information required for the basic operation. The information includes:

- Location l_0 upon initialization,
- for coverage with points b_i that define a polygon,
- (clockwise or counterclockwise) m_d ,
- Area coverage path $c_p = p, s$, parallel and spiral path planning respectively,
- Scanning radius, r_s ,
- Normal speed, s_n ,
- Current state, $state$.

Next, a signal broadcast takes place to detect the number of drones, n , currently active. When all drones are detected, a second signal is broadcast from each drone d_i which includes information about the drone's current location, l_0 , the closest, b_0 for that drone and the distance separating the two. Upon receiving border points information from the entire swarm, each drone compares all distances and retains the smallest distance from a border point b_c whether it is from a remote drone or not. This will help the swarm remain in proximity since all future calculations will take place based on this b_i .

The scanning process is linear, which means that each time the swarm is covering an area the drones move on a straight line between an initial and a which is calculated dynamically. Each initial and final point is placed on polygon edges which are separated by a third edge. For example, given a polygon q with a set of points $p_i, i = 1, 2, 3, 4$ and edges $q_j = (p_i, (p_i + 1) \bmod i_{max}), j = a, b, c, d$ an initial and a final point can exist on non-neighboring edges with a distance equal to two, for instance q_a and q_c . From now on the sides where the initial and final points are will be referred to as q_a^c and q_c^c respectively in the case of clockwise movement while q_c^r and q_a^r respectively in the case of counterclockwise.

The following step is calculating all potential , p_n –where n is the number of drones– for each drone. This process uses b_c which was retained previously. The drones check whether their movement m has been set to clockwise or counterclockwise and choose the initial polygon side q_a^c and q_c^r respectively. In the first case side $q_a^c = (b_{c-1}, b_c)$ while in the second, $q_c^r = (b_c, b_{c+1})$. All drones use b_c as the point of reference for the calculation of the first initial p_0^a – a : *initial*– and final points p_0^c – c : *final*–.

The distance separating b_c and p_0^a is equal to r_s . Each drone d_i calculates the distance from p_0^a for all drones in the swarm. In the case where its distance is the smallest, it retains p_0^a as its initial point and starts moving towards it. Otherwise, d_i updates its border points to exclude the area that will be covered by other drones.

The border points that get updated are b_c and in the case of clockwise movement b_{c+1} while in the case of counterclockwise movement b_{c-1} . The new points will be placed at a distance equal to $2r_s$ from the original points. Both points are required to change since the area between the new and old border points will be covered by another drone hence each drone will only cover one area without overlapping with the others. This process is executed continuously until each drone finds its p_0^a and p_0^c points.

The drones that reach their p_0^a first, remain in place until all drones have aligned and initialized the scanning process simultaneously. The process of altering the border points continues independently of whether the drone found its initial point first or last. This process stops based on the number of drones n . The final border points have a distance equal to nr_s from their original counterparts.

Algorithm 1: Drone Swarm Border Align for Scanning**Result:** Initial and Final Points

```

1   $l_0, b_0, d_0 = \text{euclideanDistance}(l_0, b_0),$ 
    $m_d = \{CLOCKWISE, COUNTERCLOCKWISE\}, c_p = \{p, s\}, r_s;$ 
2   $n \leftarrow \text{locateAllActiveDrones}(), \text{state} = \text{BORDER\_ALIGN};$ 
3   $\text{transmit}(l_0, b_0, d_0);$ 
4   $l_i, b_i, d_i \leftarrow \text{receiveAllRemote}(l_0, b_0, d_0), \text{addLocal}(l_0, b_0);$ 
5   $b_c, d_c;$ 
6   $b_n = 4$  #The number of border points;
7  foreach  $l_i, b_i, d_i$  do
8  |   if  $d_i < d_c$  then  $b_c = b_i;$ 
9  end
10 while  $p_0^a$  notFound do
11 |    $q_a, q_c, p_i^a, p_i^c, i \in 0 \dots b_n;$ 
12 |   if  $m_d = \text{CLOCKWISE}$  then
13 | |    $q_a \leftarrow q_c^c, q_c \leftarrow q_c^c;$ 
14 | |    $p_0^a \leftarrow b_c + r_s \in q_a, p_0^c \leftarrow b_{c+1} + r_s \in q_c;$ 
15 |   else
16 | |    $q_a \leftarrow q_c^r, q_c \leftarrow q_a^r;$ 
17 | |    $p_0^a \leftarrow b_c + r_s \in q_a, p_0^c \leftarrow b_{c-1} + r_s \in q_c;$ 
18 |   end
19 |    $d_i^l \leftarrow \text{euclideanDistance}(l_0, p_0^a)$  #Local distance from  $p_0^a;$ 
20 |   foreach  $l_i, d_i$  do
21 | |    $d_i \leftarrow \text{euclideanDistance}(l_i, p_i);$ 
22 | |   if  $d_i^l < d_i$  then
23 | | |    $\text{updateFinalBorders}();$ 
24 | | |    $\text{moveTowards}(p_0^a);$ 
25 | |   end
26 |   end
27 end
28 foreach  $l_i, d_i$  do
29 |   if  $m_d = \text{CLOCKWISE}$  then
30 | |   #Updating the border points;
31 | |    $b_c \leftarrow b_c + 2r_s \in q_a, b_{c+1} \leftarrow b_{c+1} + 2r_s \in q_c;$ 
32 |   end
33 |    $b_c \leftarrow b_c + 2r_s \in q_a, b_{c-1} \leftarrow b_{c-1} + 2r_s \in q_c;$ 
34 end

```

C.1.2 Partial Linear Coverage

By the time all steps have been completed for the self-organization and alignments process, all drones are waiting for the rest of the swarm. Upon reaching the borders, each drone broadcasts a message informing the rest of the swarm that it is ready to initiate the coverage process and changes its state to *WAITING_FOR_ALIGNMENT*. When such a message is received, the drone counts the amount it has received thus far. If the amount equals $n - 1$ and the drone has also reached the borders, it initiates the coverage process and changes the state to *SCANNING*.

Algorithm 2: Partial Area Coverage in Linear Movement

Result: Linear Coverage of an Area's Portion

```

1  $n_w = 0$  #Number of drones waiting for alignment;
2  $state = WAITING\_FOR\_ALIGNMENT$ ;
3 transmit(state);
4 while  $n_w < n$  do
5    $remote\_state = receive(state)$ ;
6   if  $remote\_state = WAITING\_FOR\_ALIGNMENT$  then
7      $n_w ++$ 
8   end
9 end
10  $state = SCANNING$ ;
11 scanUntilFinalPoint();
12  $state = WAITING\_FOR\_ALIGNMENT$ ;
13 transmit(state);
14  $n_w = 0$ ;
15 while  $n_w < n$  do
16    $remote\_state = receive(state)$ ;
17   if  $remote\_state = WAITING\_FOR\_ALIGNMENT$  then
18      $n_w ++$ ;
19   end
20 end
21  $state = BORDER\_ALIGN$ 

```

The route each drone covers is linear. It begins from the initial point p_a^c , which is calculated and reached during the alignment process. The destination is the final point p_c^c , also calculated during the alignment process. Each drone covers an area of size equal to the distance between p_a^c and p_c^c multiplied by the predefined scanning radius r_s .

When a drone reaches the final point p_c^c it informs the rest of the swarm that it has finished and changes its state to `WAITING_FOR_ALIGNMENT`. When all drones reach their final point the state of each drone in the swarm changes to `BORDER_ALIGN`. Despite all swarm drones moving in parallel, variable conditions such as wind and changes in speed (more details in subsection C.2.3) might affect the destination arrival time. By informing the rest of the drones, the swarm can initiate the coverage processes simultaneously.

During the latter steps of the initial and final points calculation process, each drone updates the border points to find the area that will remain after the scanning takes place by the entire swarm. Thus, when the coverage finishes, the drones are outside of the new area –or polygon– that has yet to be covered. The entire process is described in Algorithm 2.

The processes described in Algorithms 1,2 are executed multiple times until the area of interest is fully covered. The area coverage path is altered by manipulating the movement direction m_d . For instance, if m_d remains unchanged between linear scans the result will be a spiral-like coverage shape, while alternating it between linear coverage paths can produce a back-and-forth –or parallel lines– path. The following subsection describes the process of information collection during the swarm’s scanning period.

C.2 Collecting Information from Points of Interest

During the scanning period, the drones’ main purpose is to and that describe their current state. The algorithm states that each drone has a specific scanning radius that covers an area of size $r_s^2\pi$. When a PoI is located within the scanning radius of a drone, the latter takes specific actions to collect as much information as possible.

When a PoI enters a drone’s scanning radius, measurements are performed to

determine whether speed alteration is required or not. The measurements include:

- The distance (in meters) between the drone and the PoI d_l^p ,
- The distance (in meters) for which the PoI will remain within the scanning radius d_s ,
- The time required t_s for the data collection,
- The new drone speed.

C.2.1 Distance Between Drone and Point of Interest

Distance between a drone and the PoI can be measured with various means which are based on the application-specific requirements and sensors. For instance, in applications where a forest area needs to be scanned for possible fire points infrared cameras are employed to detect locations of increased temperature. In this case, *Euclidean Distance* can be used by employing the camera's pixel information and by assuming that the drone is located at the center of each photograph.

In a scenario where the drone swarm is required to locate wireless transmissions from a WSN that is set up in a geographical area, the location of the nodes can be derived by using signal triangulation methods. In the context of the current thesis, the distance d_s is derived using the Euclidean formula for 2-dimensional planes. The locations of both the drone and the PoI are retained by the simulation software as points on a canvas.

C.2.2 Distance within Radius

The drone has a predefined speed, s_n which is provided during the initialization process. This speed cannot be exceeded. In the case where a drone requires information from a PoI and the current speed is preventing the complete collection due to small distance d_s , the speed can be reduced to meet the specific requirements. The new speed is calculated based on the distance d_s .

Figure C.1 depicts a drone (the black dot in the center of the scanning radius circle) with a scanning radius $r_s = 50m$. The drone's normal movement speed is $s_n = 4m/s$ or $s_n = 14.4km/h$ with a direction to the right as depicted by the arrow on top. It is assumed that the drone's moving direction is parallel to the x-axis.

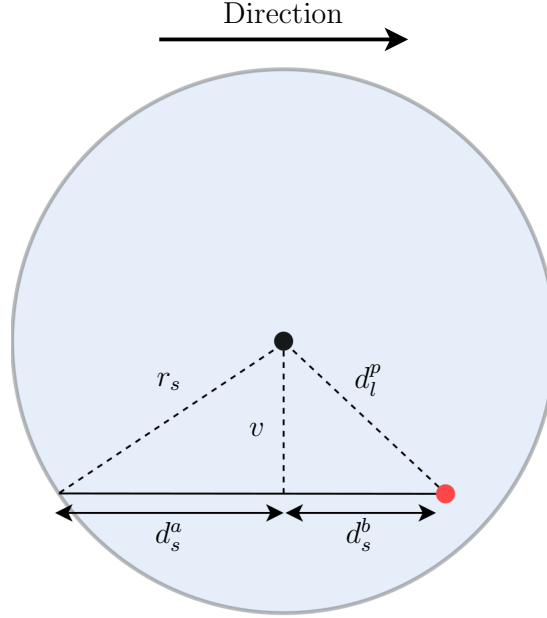


Figure C.1: Locating a PoI and calculating the distance in scanning radius d_s .

The drone assumes that it is within a grid of x- and y-axes where its location is at the center or point $(x = 0, y = 0)$. On the bottom right corner, a PoI (red dot) has been located by the drone. To measure the distance d_s the drone creates two separate triangles where each has a 90° corner.

The summation of d_s^a and d_s^b sides from the separate triangles equals d_s . The distance d_l^p is calculated using the Euclidean distance formula in 2-dimensional spaces. Edge v is equal to the distance between the drone and the PoI in the y-axis, while d_s^b is equal to their distance in the x-axis.

Moving on to the triangle on the left, it is also an orthogonal triangle with two known sides v and r_s . The aim is to obtain the final edge d_s^a to gain knowledge relative to the distance the PoI will remain within the radius. By employing the *Pythagorean Theorem* using the already known triangle sides distance d_s^a can be calculated. Finally, distance d_s can be obtained by adding up the smaller distances d_s^a and d_s^b .

C.2.3 Speed Adjustment

The algorithm states that the swarm has a predefined upper speed limit. In the case where this speed remained unchanged during a mission, the time required to complete it could be calculated purely based on the required travel distance. To avoid any information loss, the current algorithm states that a drone can decrease its speed based on an estimation of the amount of information $-i_p$, information from PoI, measured in bits– it will require from a newly located PoI. The estimate can vary based on the sensors and the application-specific requirements.

Each drone uses i_p and a rate i_r at which the information will be gathered to calculate the required amount of time:

$$t_p \leftarrow i_p / i_r \quad (\text{C.1})$$

The next step is to alter the flying speed based on distance d_s , time t_p and an additional slowing down percentage per_s :

$$speed \leftarrow per_s(d_s/t_p) \quad (\text{C.2})$$

The percentage will provide the drone with an additional small time window which will account for any speed alterations due to external factors (such as wind).

The final step is to inform the rest of the swam about the speed change and the amount of time this speed will last for. The swarm will adjust its speed to remain in the original formation. In the case where multiple speed changes have occurred, each drone sets the speed based on the slowest value it has detected. All speed values are retained and speed duration is updated at all times independently of whether it is the current speed or not. The speed cannot exceed the originally provided amount at any time.

Chapter D

Simulation Results

THE algorithm's effectiveness is evaluated within a simulated environment. This process involves the programming and development of the appropriate mechanics that provide the basic operation of a swarm. Also, this process creates the foundation where the current algorithm and the final drone behavior can be examined thoroughly. The development is conducted using the programming language C++ and the software hosting the environment is OMNeT++ at version 5.6.2.

The following sections detail the simulations setup (Section D.1) as well as the results that accrue from the simulations (Section D.2).

D.1 Simulations Setup

The algorithm in its core has many variables one can examine. A canvas is created within the OMNeT++ simulation which includes various components. Figures D.2,D.3,D.4 depict the environment that is developed. On the top left corner, a component that handles the animations is included named UAVAnimator. Moving further down, a polygon is depicted with white borders which are defined by four points. This is the area the drones are covering. The area's size is 800×800 or $640.000m^2$. The scale of the map is 1:1. This means that if a drone is moving along the x-axis for each 1 value added to its x variable, it has traveled 1 meter.

All initial points on the map are randomly selected by the uniform distribution. Two *Random Number Generators (RNGs)* are used as different seed sets, one for the

initial location of the drones and one for the PoIs. This helps keeping independent locations between simulations for the drones and the PoIs. For instance, in the case where a simulation is executed for different swarm sizes but the same number of PoIs are placed on the map, the latter will have to retain the same locations between simulations in order to derive more accurate results.

For the evaluation of the algorithm the following swarm variables are used:

- Scanning radius $r_s = 25, 50, 100$
- Number of drones in the swarm $n = 1, 2, 4$.
- Number of PoIs, equal to 25, 50, 100, 150, 200, 250, 300.
- Repetition number, which effects the random number generation by providing different seed set numbers. A total of 8 repetitions are examined.
- Area coverage path, two different paths are used $c_p = p, s$ where p : *parallel* and s : *spiral*.

The total number of simulations is equal to 1008. When the area coverage is complete, each drone exports one line into a file which includes the following metrics:

- Time of coverage, it includes the duration for which the swarm either covers/scans the area or changes formation to reenter the new area.
- The number of slowdowns.
- The mean speed during the entire simulation.
- The percentage of PoIs for which drones collected information.
- Total distance traveled.
- The number of switched formations.

The repetition numbers only change the locations of the PoIs. For instance, if the number of PoIs is equal to 200, $n = 2$, $r_s = 50$, and $c_p = p$ then the only variable that is altered is the location of the PoIs. This helps gain more consistent results since the edge cases of the algorithm can be smoothened out.

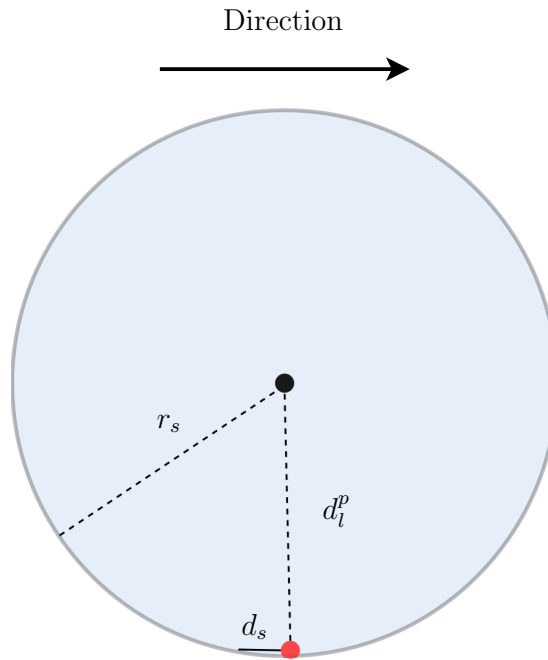


Figure D.1: A drone covering an area and discovering a PoI at the most remote point of its scanning radius.

Edge cases are detected during the information collection time and they mainly affect the total area. Figure D.1 depicts an edge case where a drone scanning an area discovers a PoI which is located at the most remote area of its scanning radius. When a drone discovers such a PoI the algorithm is coordinating the drone towards collecting as much information as possible, thus the drone has to drastically reduce its speed, a step which all swarm drones follow.

Figures D.2, D.3, and D.4 depict the simulated environment where 1, 2, and 4 drones cover the area respectively. Points of interest are spread though the area and the effect of two RNGs is observed. All PoIs have the same locations in all three figures. The drones are connected in a full graph and the effect long distance wireless communications is not taken into account.

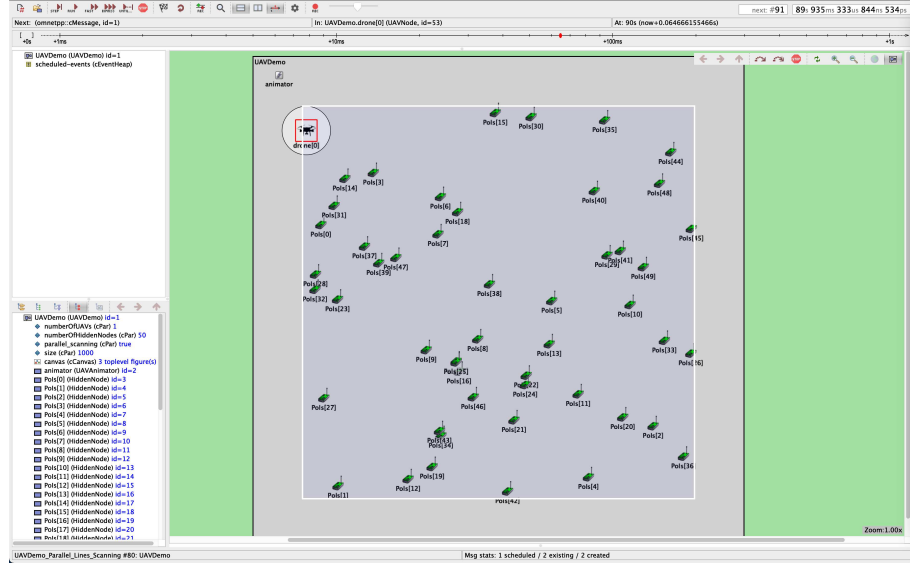


Figure D.2: The simulated environment in OMNeT++ where 1 drone with 50 meters of scanning radius is depicted. The area of interest is within the white borders where points of interest are also depicted.

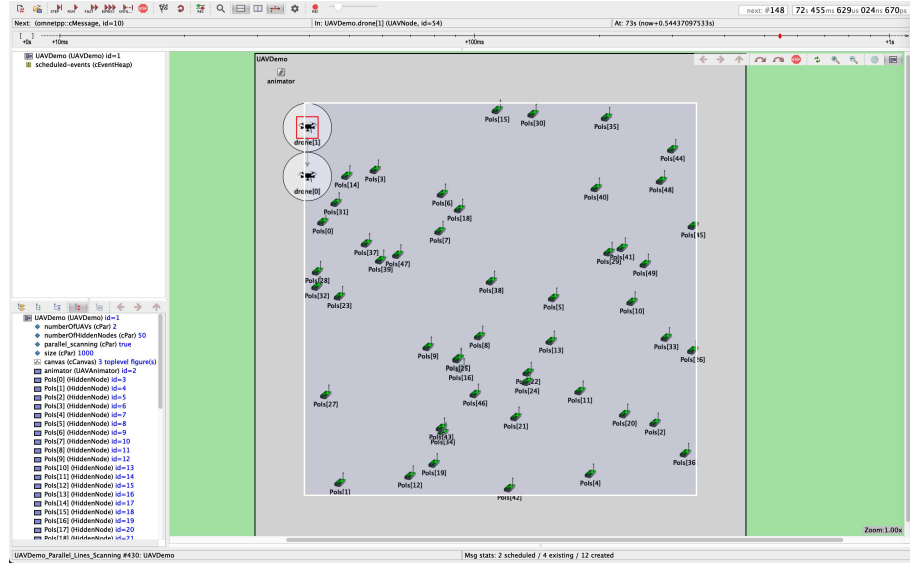


Figure D.3: The simulated environment in OMNeT++ where 2 drone with 50 meters of scanning radius is depicted. The area of interest is within the white borders where points of interest are also depicted.

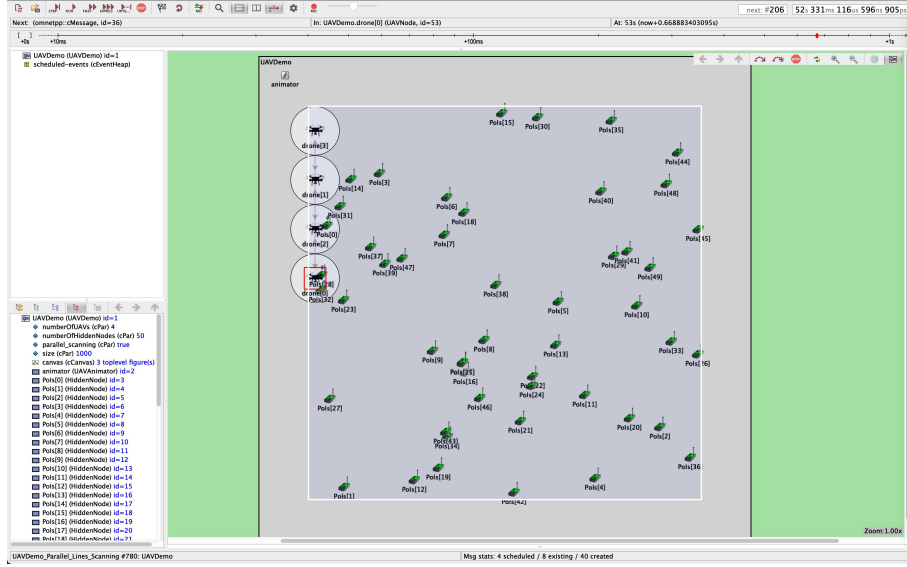


Figure D.4: The simulated environment in OMNeT++ where 4 drone with 50 meters of scanning radius is depicted. The area of interest is within the white borders where points of interest are also depicted.

D.2 Performance Evaluation

The results that accrue from the simulations showcase the effectiveness of the proposed algorithm.

D.2.1 Area Coverage Path

Careful examination of the results shows that an area covered by the swarm can differ in terms of required travel distance and time. An important factor affecting the two metrics is the area coverage path. Two different paths are examined, parallel lines and spiral. Figure D.5 depicts the paths each coverage path creates.

The travel distance between the two shapes differs because each has a variable number of switched formations. For example, the travel distance when the scanning radius is equal to 25 for the parallel lines is 13524m while for the spiral is 13918m. The difference between these numbers might affect the swarm's ability to cover an area. The swarm has to switch formation between partial linear coverage which leads to excess travel distance since for every formation switch, the drones have to

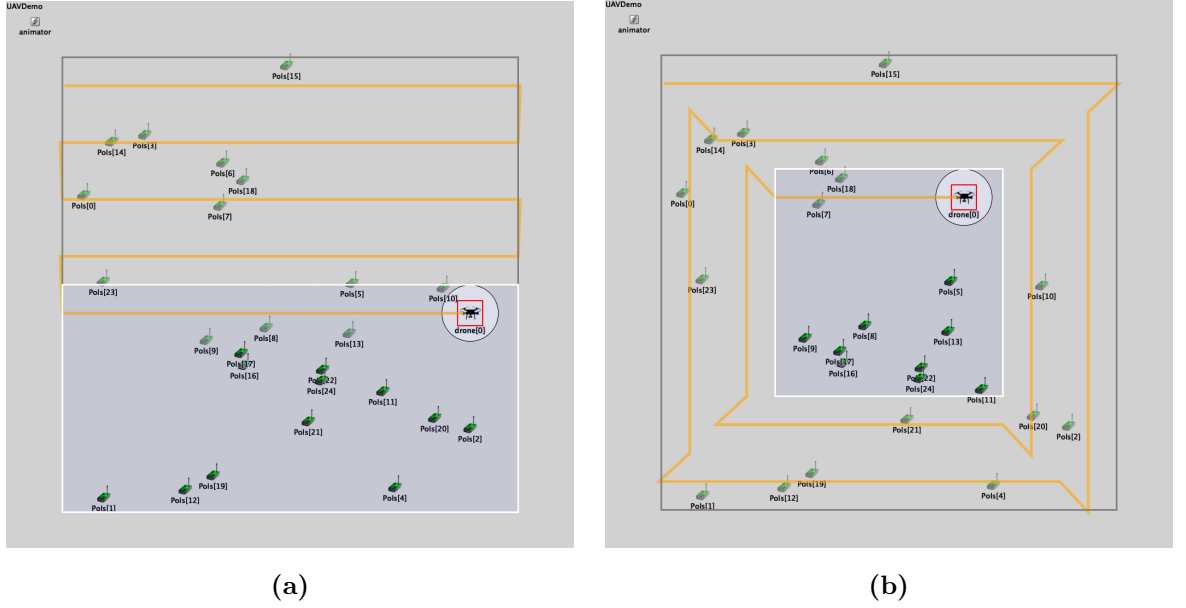


Figure D.5: Examples of parallel lines and spiral area coverage. The yellow line depicts the path of the drone: **(a)** Parallel lines area coverage. **(b)** Spiral area coverage.

re-enter the area of interest, hence they fly outside the borders.

The distance outside the borders d_o can be estimated for both coverage paths. For the parallel lines this distance is:

$$d_o^p = 2r_s f p_a^c, \quad (D.1)$$

where p_a^c is the number of initial points to check until the closest one is found, and f is the number of formation switched until the area's full coverage. The spiral shape is a bit more intricate since the new formation leads the drones at obtaining locations that form a linear path that is parallel to the original line of movement or to the previous formation.

The distance outside the borders for a drone covering an area with a spiral shape is:

$$d_o^s = f \sqrt{(d_b^l)^2 + (2r_s(p_a^c - 1) + r_s)^2}. \quad (D.2)$$

The d_b^l is the distance from the closest border point. The first initial point that is calculated has a distance of r_s from the drone while the rest of the initial points

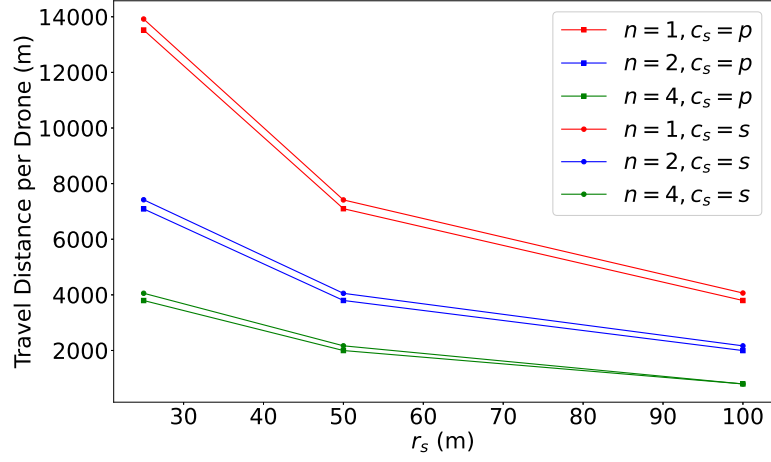
have a distance $r_s(p_a^c - 1) + r_s$. These distances constitute an orthogonal triangle and its hypotenuse is d_o .

By comparing the two shapes it is derived that parallel lines provide less travel distance than spiral. Figure D.6 depicts the travel distance for a drone with scanning radius $r_s = 25, 50, 100$. The lines with the squared points represent the parallel shape and the round ones depict the spiral shape. Every time the number of drones is cut in half the travel distance for the same r_s is doubled which is quite reasonable since the size of the area that requires coverage remains the same. This is true for both shapes.

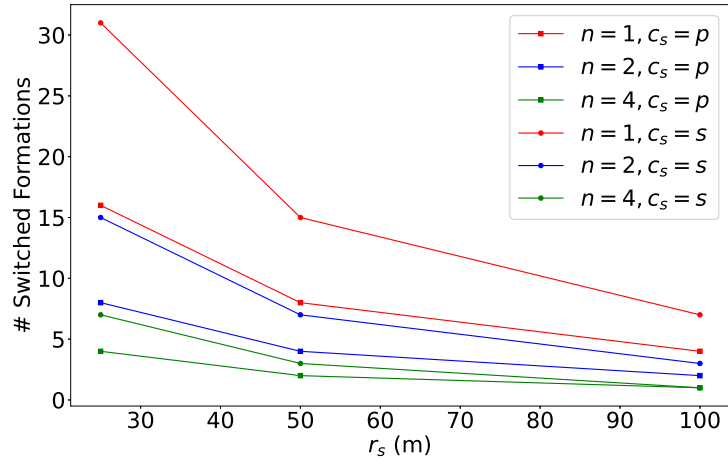
The number of switched formations for scanning radius $r_s = 25$, $n = 1$ and coverage path $c_p = p$ is equal to 16 while for $c_p = s$ is equal to 31. Based on the equations D.1, D.2 for the calculation of outside the border distance, d_o is calculated for both shapes for the scenario described above. For the parallel shape, the d_o is 800 meters while for the spiral d_o is equal to 1096 meters. The equations are proven since if this distance is subtracted by the equivalent total coverage distance for each shape with $r_s = 25$ and $n = 1$, the result is 12724 and 12821 meters respectively. These are the meters that the drone has traveled to cover the entire area inside the borders.

These numbers do not match which accounts for inaccuracies during the simulations. The graphics engine on OMNeT++ required the development of custom functions for detecting when a moving object reaches a point. When the drone reaches the final point in the current implementation, their window of distance is between -4 and 4 which can explain the drifting of the distances. These distances should be the same based on the algorithm since in both cases the drones have to cover the same area using bigger trails in the case of parallel while using smaller multiples for the spiral.

Similar behavior is observed throughout all the simulations for 25, 50, and 100 scanning radius. In more detail, the number of switched formations for all simulations in spiral area coverage is almost double compared to the parallel lines coverage path. The only case where this is not true is for scanning radius $r_s = 100$ and $n = 4$ where the swarm covers the entire area with one linear movement. The swarm does not have to switch formation since the diameter of coverage for each drone is equal to $2r_s = 200$ and multiplied by the number of drones it is well understood that the entire area (800×800) is covered in one move.



(a)



(b)

Figure D.6: The mean distance a drone traveled during the simulations for a swarm size equal to 1, 2 and 4 drones based on the scanning radius r_s : **(a)** Depicts the mean distance per drone while the r_s changes in size. **(b)** The number of switched formations for parallel and spiral coverage paths based on variable r_s .

Careful examination of the distance data can provide an insight as to which area coverage paths can benefit the swarm's operation. For instance, a shape with small

edges or a coverage path that results in the creation of smaller edges of the original area can potentially lead to unnecessary consumption of resources. This can be observed by examining the results for the two coverage paths from the simulations. In the case of spiral area coverage, each time the drones linearly covered a portion of the area, they created smaller edges. This resulted in requiring almost a double number of formation switches and each time the swarm attempts to reenter the area, it flies over an area that was previously covered and is currently outside the new borders forcing it to consume additional time and resources.

D.2.2 Variable Scanning Radius Values

The results derived by these simulations are depicted in the Figures D.7, D.8, D.9, D.10, D.11, and D.12. The results for three different scanning radius values, $r_s = 25, 50, 100$, and two area coverage paths $c_p = p, s$, parallel and spiral respectively are examined. Each Figure comprises four (4) subfigures. The metrics are relative to the PoIs per simulation. Subfigure (a) depicts the mean time required to completely cover the area. Subfigure (b) depicts the mean number of slowdowns required to scan as many of the PoIs as possible. Subfigure (c) depicts the mean speed for each drone in the swarm during each simulation. Subfigure (d) depicts the mean percentage of successfully scanned PoIs.

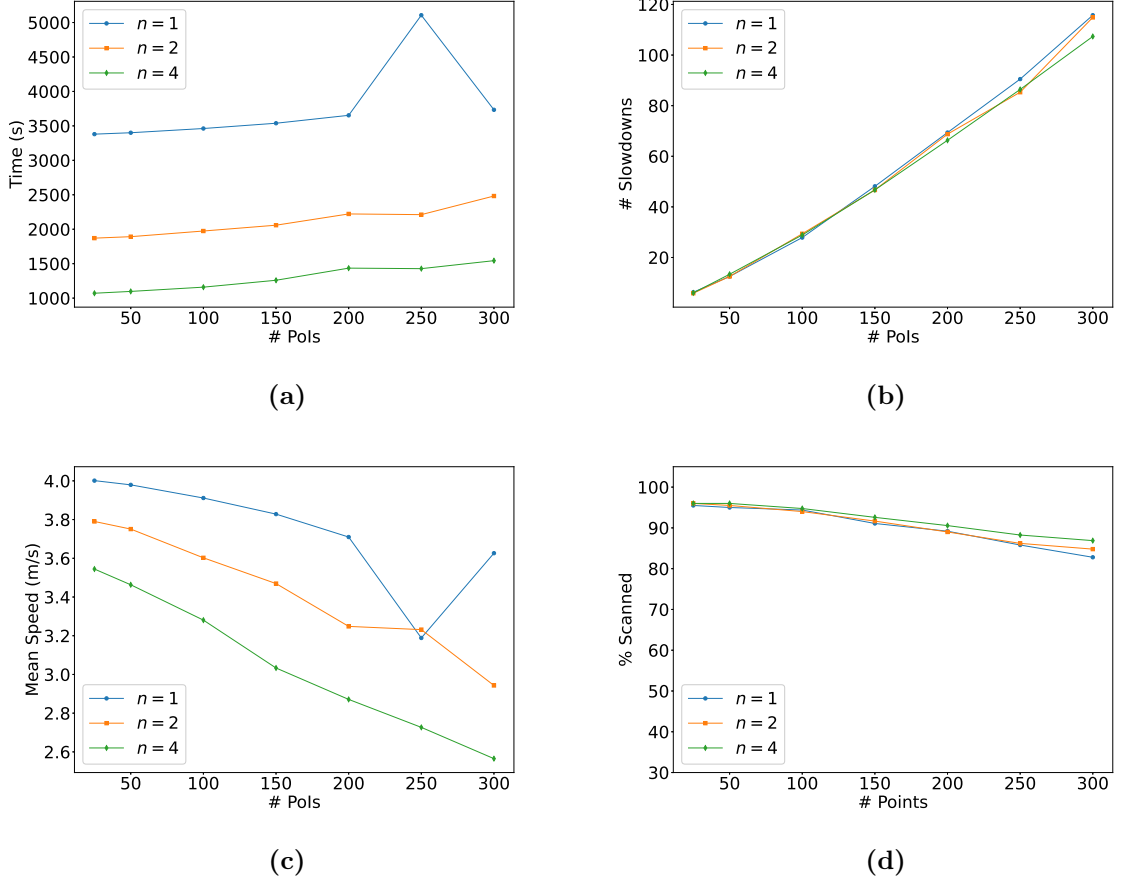


Figure D.7: Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 25m$ and parallel lines coverage path: **(a)** Time required for the complete area coverage by the initial number of PoIs. **(b)** The number of slowdowns required for the information collection from the PoIs. Depends on the number of PoIs that exist inside the map. **(c)** Mean speed throughout the simulation with variable number of initial PoIs. Depends on the number of slowdowns. **(d)** The percentage of scanned PoIs while they increase.

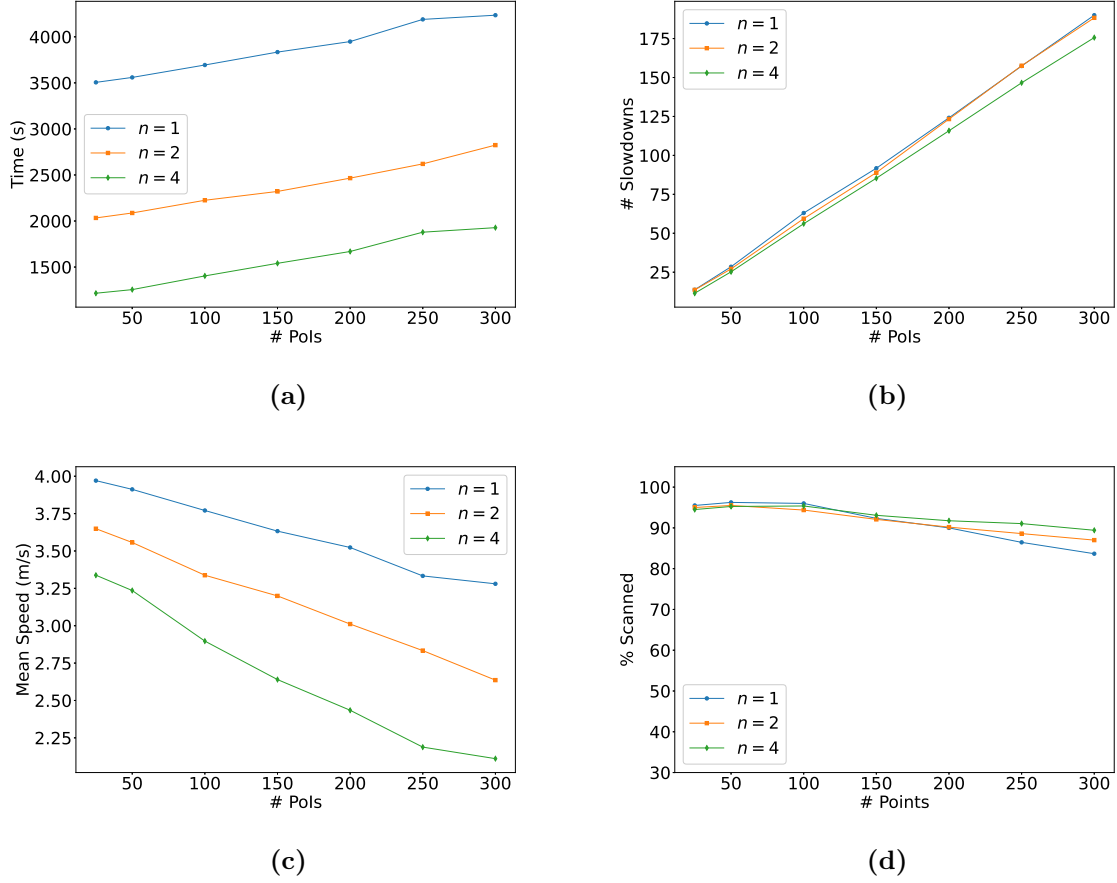


Figure D.8: Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 25m$ and spiral coverage path: **(a)** Time required for the complete area coverage by the initial number of PoIs. **(b)** The number of slowdowns required for the information collection from the PoIs. Depends on the number of PoIs that exist inside the map. **(c)** Mean speed throughout the simulation with variable number of initial PoIs. Depends on the number of slowdowns. **(d)** The percentage of scanned PoIs while they increase.

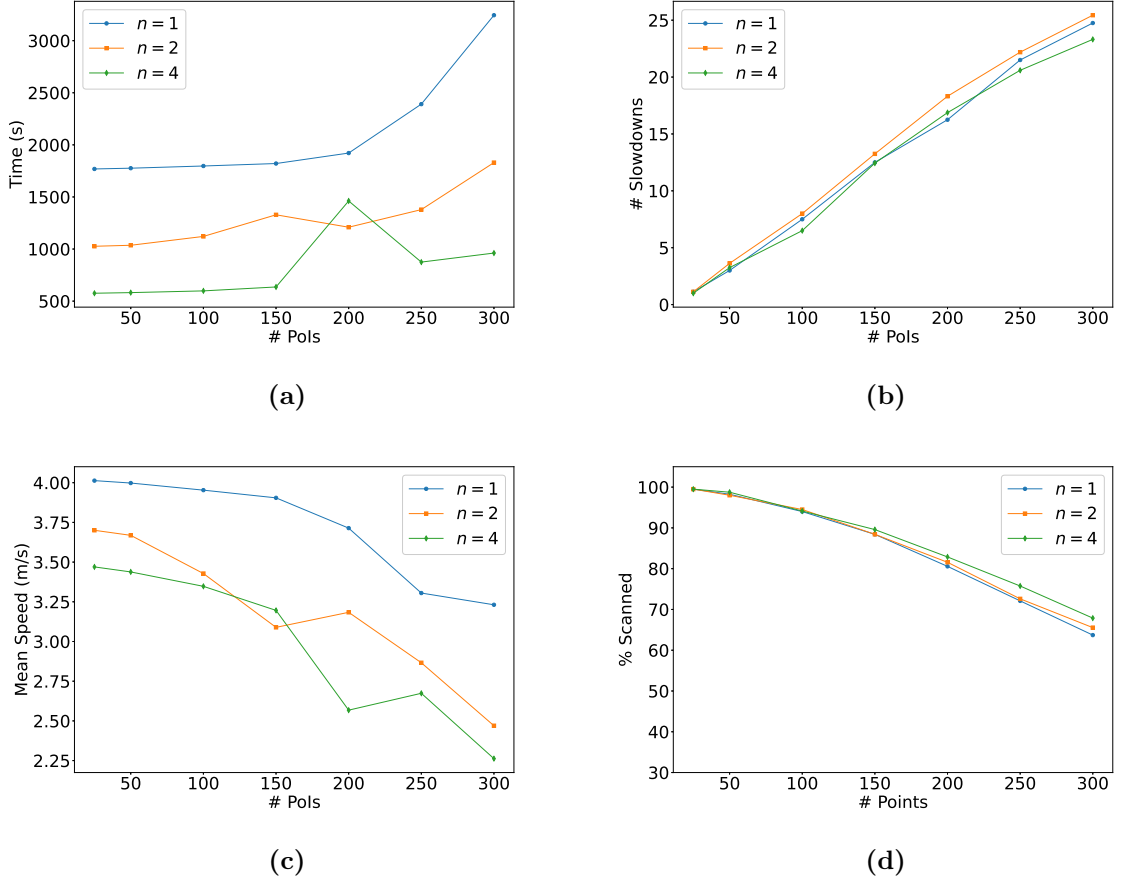


Figure D.9: Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 50m$ and parallel lines coverage path: **(a)** Time required for the complete area coverage by the initial number of PoIs. **(b)** The number of slowdowns required for the information collection from the PoIs. Depends on the number of PoIs that exist inside the map. **(c)** Mean speed throughout the simulations with variable number of initial PoIs. Depends on the number of slowdowns. **(d)** The percentage of scanned PoIs while they increase.

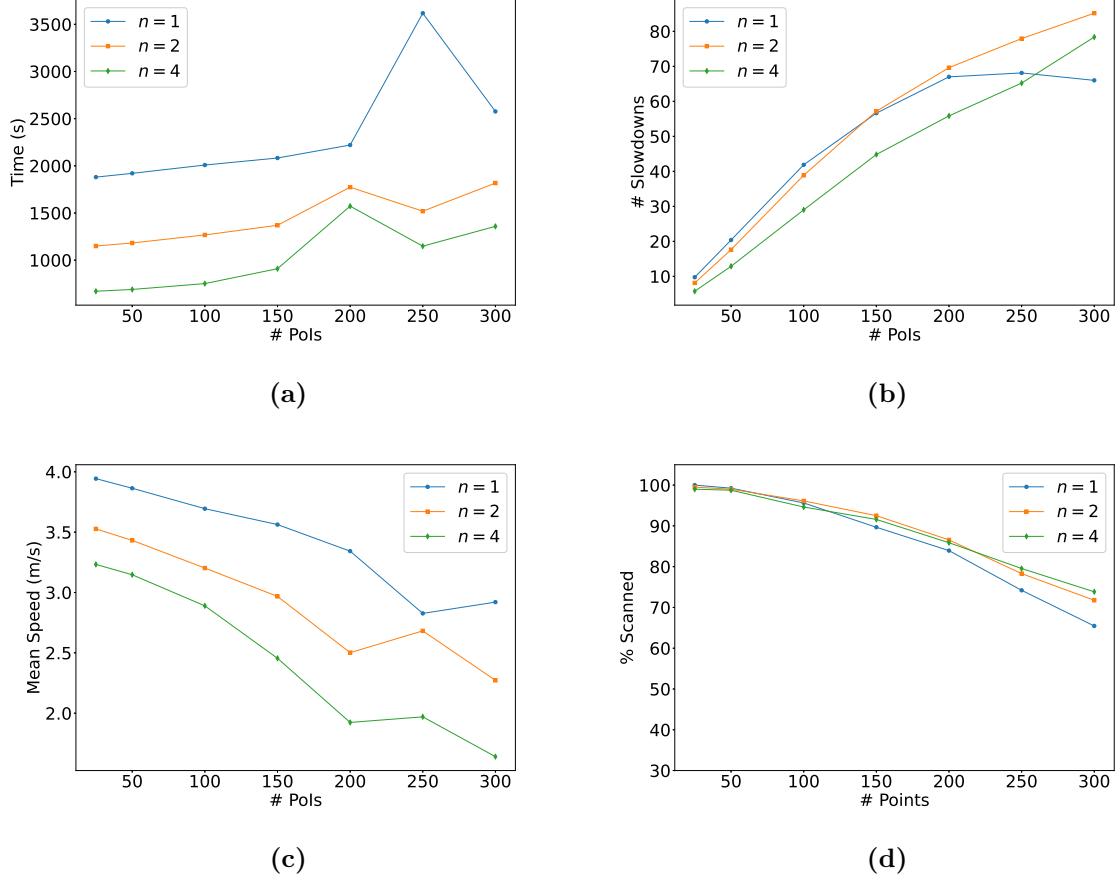


Figure D.10: Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 50m$ and spiral coverage path: **(a)** Time required for the complete area coverage by the initial number of PoIs. **(b)** The number of slowdowns required for the information collection from the PoIs. Depends on the number of PoIs that exist inside the map. **(c)** Mean speed throughout the simulations with variable number of initial PoIs. Depends on the number of slowdowns. **(d)** The percentage of scanned PoIs while they increase.

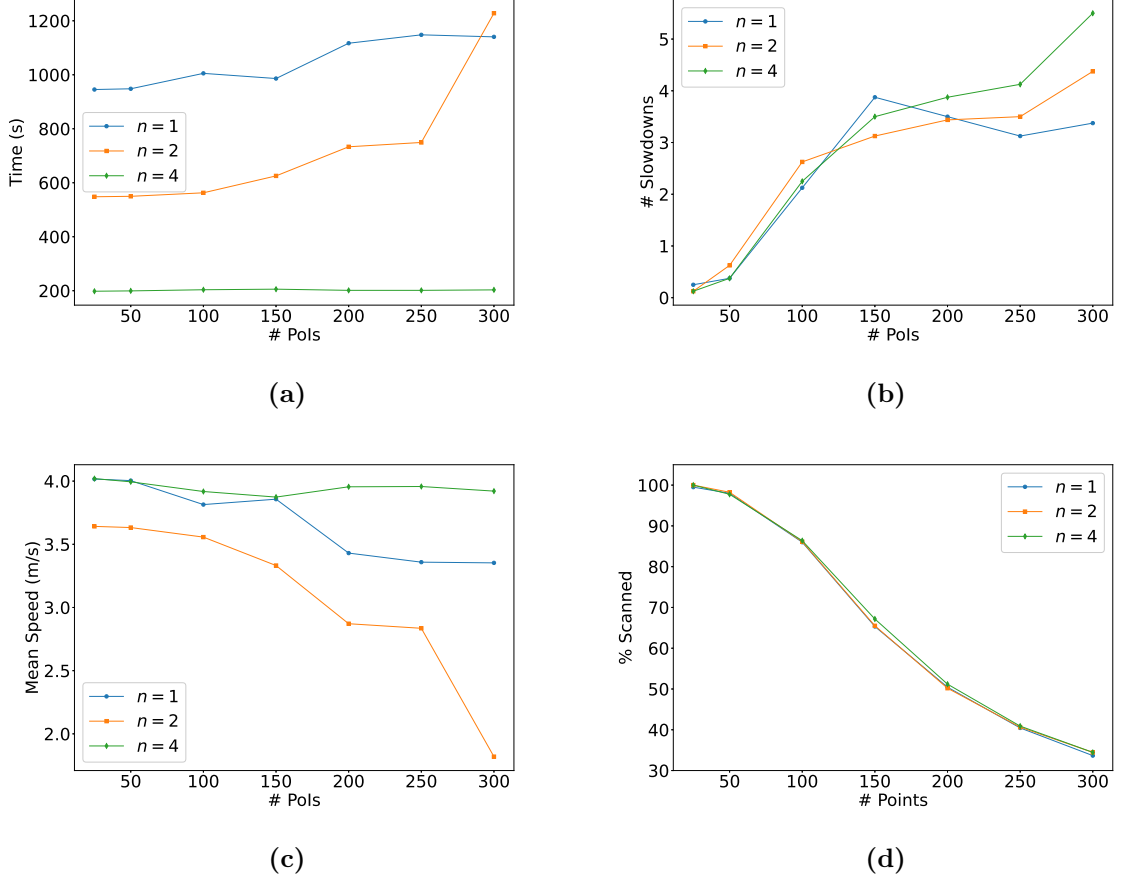


Figure D.11: Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 100m$ and parallel lines coverage path: **(a)** Time required for the complete area coverage by the initial number of PoIs. **(b)** The number of slowdowns required for the information collection from the PoIs. Depends on the number of PoIs that exist inside the map. **(c)** Mean speed throughout the simulations with variable number of initial PoIs. Depends on the number of slowdowns. **(d)** The percentage of scanned PoIs while they increase.

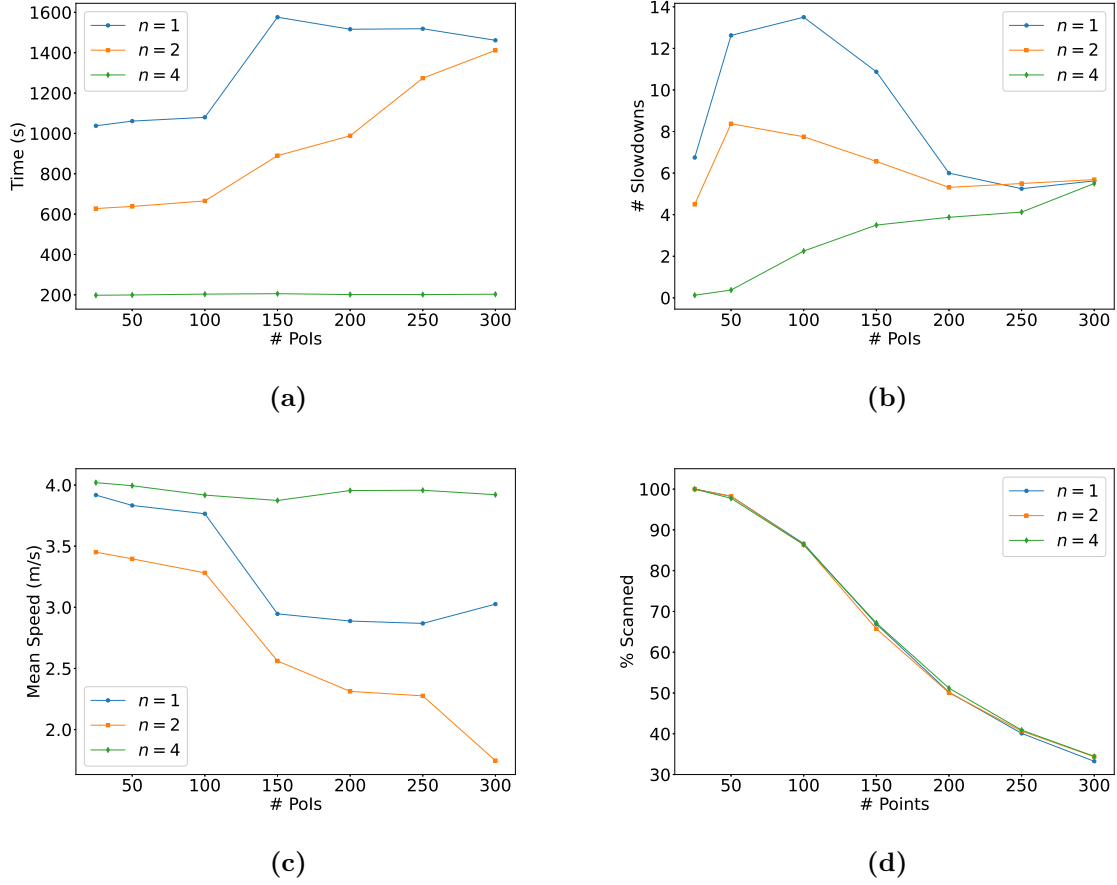


Figure D.12: Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 100m$ and spiral coverage path: **(a)** Time required for the complete area coverage by the initial number of PoIs. **(b)** The number of slowdowns required for the information collection from the PoIs. Depends on the number of PoIs that exist inside the map. **(c)** Mean speed throughout the simulations with variable number of initial PoIs. Depends on the number of slowdowns. **(d)** The percentage of scanned PoIs while they increase.

Subfigure (a) from each figure depicts the diagram of time. Based on the distance analysis in both coverage paths, in Subsection D.2.1, similar behavior is depicted with the coverage time. More specifically, coverage time for a swarm with parallel lines is always smaller compared to spiral coverage. The analysis that takes into

account the distance outside the borders b_o can explain the reason for such a time difference. Time percentage changes are depicted in Tables D.1,D.2. Each time the number of drones is doubled the coverage time for the same area is decreased by half. The time percentage increase between the two coverage paths is observed due to the number of swarm changes or border aligns (i.e the drones are outside the area) is almost double as well.

Tables D.1, and D.2 depict the time increases for a swarm with 1, 2, and 4 drones between the spiral and parallel coverage paths. Two cases with 25 and 100 PoIs are selected. It is observed that the time percentage increase is larger for the spiral coverage path in all cases and they increase as more PoIs are added. The mean time for $r_s = 100$, $n = 4$ equals zero since the swarm covers the area in one linear move without switching formation in both coverage paths.

Time percentage increase in all cases where the scanning radius is $r_s = 100$ and the number of drones is $n = 4$ is equal to zero since the swarm in both parallel and spiral shapes require only one border align to fully cover the entire area. The mean speed that is measured seems to be a mirror of the time required for the area completion in all curves. This is attributed to the fact that mean speed is calculated based on the distance traveled and the time required for a drone to fully cover the area. The area throughout the simulations remains unchanged in size.

On few occasions, it is observed that spikes occur in the otherwise normal evolution of the curve. For instance, in Figure D.10 the time for one (1) drone to completely cover the area where 250 PoIs exist as well as the time for two (2) and four (4) drones in an area with 200 PoIs depict spikes. These occur when a drone discovers a PoI in such a place that is considered an edge case.

Details relative to the number of slowdowns required for the discovery of PoIs are depicted in all subfigures (b). The number of slowdowns is affected by the scanning radius. As the scanning radius is increased, the number of slowdowns is decreased. This occurs since more PoIs exist within the same scanning radius, hence more PoIs enter and exit the radius for the same travel distance while the drone only collects information from one of them. Comparison between the two coverage paths shows that the spiral shape requires more slowdowns for the same percentage of successfully scanned PoIs.

As the scanning radius is increased, the behavior between different numbers of drones for the swarm also changes compared to smaller values. More specifically,

r_s	n	Parallel Coverage	Spiral Coverage	Percentage Increase
25	1	3380.0	3505.5	+3.58%
25	2	1870.625	2033.375	+8.0%
25	4	1071.0	1215.75	+11.91%
50	1	1768.75	1880.125	+5.92%
50	2	1026.5	1149.625	+10.71%
50	4	575.75	670.5	+14.13%
100	1	945.375	1037.75	+8.9%
100	2	548.125	627.75	+12.68%
100	4	198.125	198.125	+0.0%

Table D.1: Percentage increases between Parallel and Spiral coverage paths in Time for 25 PoIs.

when a drone in a swarm with more than one (1) requires that its speed is decreased, the entire swarm has to follow it. This provides time for remote drones to discover more PoIs since they remain within the same area for longer periods. As a result, the successfully scanned percentages are increased compared to a single drone even though the number of slowdowns might be smaller.

When the scanning radius has the maximum value, $r_s = 100$, the drones have more PoIs within the scanning radius but only collect information from one at a time. The movement of the drone only decreases when a PoI is detected from which the drone requires excessive information. This is why major percentage decreases are observed for $r_s = 100$ while the number of PoIs increases. The spiral shape also seems to affect the number of slowdowns since the perspective, i.e. the flying direction, of the drone changes for the same area. This means that when a drone enters an area where 20 PoIs exist from different sides, it will have a variable number of PoIs closest to it from each side.

Based on the presented results, it is derived that parallel lines scanning is more efficient for all metrics for an area with the current characteristics. Also, for a smaller scanning radius, the swarm can collect information from more PoIs compared to larger scanning radius values.

r_s	n	Parallel Coverage	Spiral Coverage	Percentage Increase
25	1	3461.75	3694.0	+6.29%
25	2	1974.0	2225.125	+11.29%
25	4	1159.25	1403.75	+17.42%
50	1	1797.125	2008.25	+10.51%
50	2	1121.0	1267.125	+11.53%
50	4	598.25	751.5	+20.39%
100	1	1005.25	1079.75	+6.9%
100	2	562.75	665.375	+15.42%
100	4	203.625	203.625	+0.0%

Table D.2: Percentage increases between Parallel and Spiral coverage paths in Time for 100 PoIs.

D.2.3 Data Collected

The algorithm states that when a drone discovers a PoI it calculates the distance for which it will remain within its scanning radius. This is based on the movement direction and the distance from the drone. Also, drones estimate the amount of data that will be collected from each PoI. A data rate is used to determine the required time for the data collection from a PoI. The data rate depends on the application and the ability of the employed sensors to collect/measure information from the surrounding environment.

In the current simulations, an effort is made to generalize all algorithmic processes in order to allow it to operate in various applications. To obtain the simulated results, a data rate of 250.000 bytes per second (250kBps) is used. This number is 2 mega bits per second and it is chosen in order to represent applications which use imaging as sensors for the data collection where a single image is taken per second and analyzed in order to provide a less energy-consuming process. In a real-life mission where cameras are used to measure an environmental attribute, they require to capture images from multiple perspectives to obtain more accurate information. This can be achieved by performing a periodic image capturing while the drone is constantly moving independently of the current speed.

Upon discovery of a new PoI the drone estimates the amount of data that will

be required for collection. A random integer number is chosen from a uniform distribution between 1 and 2 megabytes that represents the amount of the required information for a specific PoI. The drone then calculates the amount of required time for the collection and adjusts the speed accordingly. The swarm also reduces its speed when required.

Figures D.13, D.14, D.15, D.16, D.17, and D.18 depict the data collected through the experiments. All figures comprise two (2) subfigures, (a) depicts the mean data collected per drone, and (b) depicts the total data collected by the swarm during the simulations. The figures are grouped based on the scanning radius and the total number of PoIs in the area of interest is used for the x-axis. Each curve represents a swarm with 1, 2, and 4 drones respectively. Two groups have been included for each scanning radius, one for the parallel lines coverage path and one for the spiral.

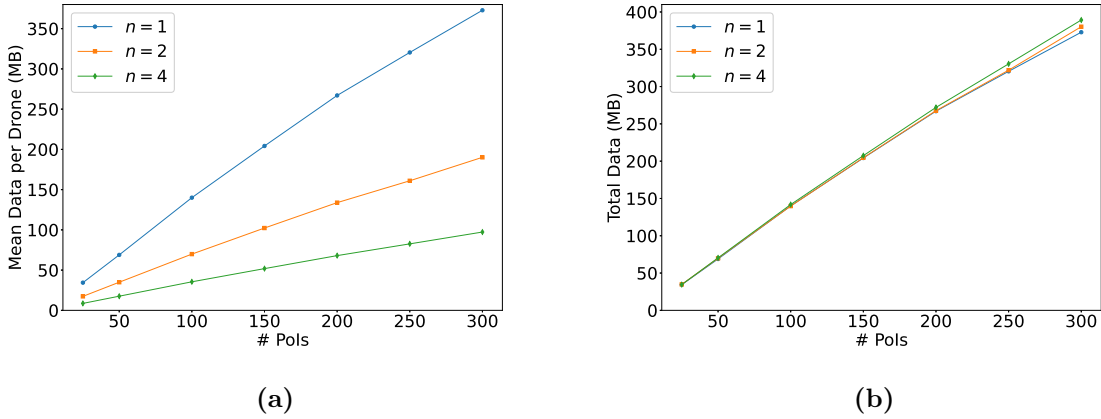


Figure D.13: Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 25m$ and parallel lines coverage path. The figures depict metric relative to the data collected by the drones during the mission. The data is measured in MegaBytes (MB): **(a)** Mean data per drone for missions with variable number of PoIs. **(b)** Total data collected by the swarm.

By examining the first subfigure (a) the main observation is that while more drones are added to the swarm, the workload for each individual is reduced by almost half. The swarm with a single drone is depicted having the most intensive workload and this is apparent when combining the time data detailed in Subsection D.2.2.

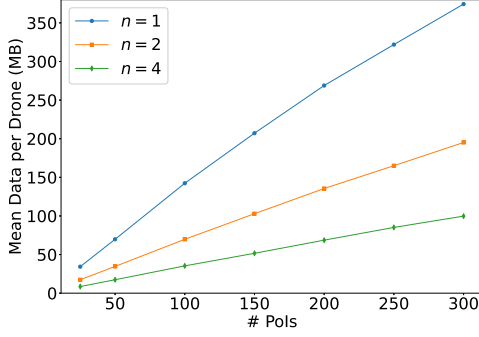
The curves of time and mean data collected per drone depict similarities which showcase the workload required for swarms with variable number of drones. Time also decreased by almost half each time the drones were doubled.

Slight variations in collected data appear between the spiral and the parallel coverage paths for each scanning radius value which are not immediately apparent in the subfigures. They are attributed to the random number of data selected for each PoI and thus are considered insignificant. Comparison between the numbers of scanning radius reveals that each time it is doubled, the swarm's ability to collect information deteriorates and this is true for both coverage shapes. More specifically, for scanning radius equal to 25 meters the peak data collected by the swarm is equal to 399 megabytes, for 50 meters the same value equals 329, while in the 100 meter scanning radius the swarm collected only 153 megabytes.

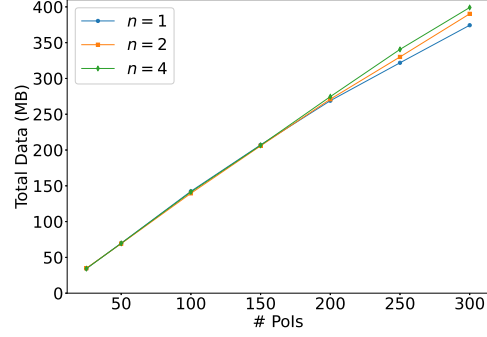
The collected data displays the same behavior as the number of slowdowns. This happens due to the fact that slowdowns occur only when a drone requires more time to collect information from a PoI in the area of interest hence the collected data distribute similar behavior. Also, when the swarm has more than one (1) drones, slowdowns, which happen at the same time in the entire swarm, benefit the data collection in remote drones since, in the case where multiple PoIs exist in the area where the slowdown takes place, they are provided with additional time, hence they can collect more information.

The total number of data collected by the swarm reveals differences between the parallel lines coverage and spiral paths. In both cases, the swarms with more drones can collect more data. This is attributed to the slowdowns which happen for the entire swarm independently of whether all drones require more time or not. For smaller numbers of scanning radius, both shapes exhibit similar data collection numbers with a slight increase in the spiral shape for swarm sizes two (2) and four (4).

For increased scanning radius –50 meters–, differences in data collection between swarms with a single drone and multiple are more apparent especially in the case of spiral coverage. When the scanning radius reaches its maximum value –100 meters–, all swarms collect the same amount of data for different numbers of drones. This is also true for both coverage shapes.

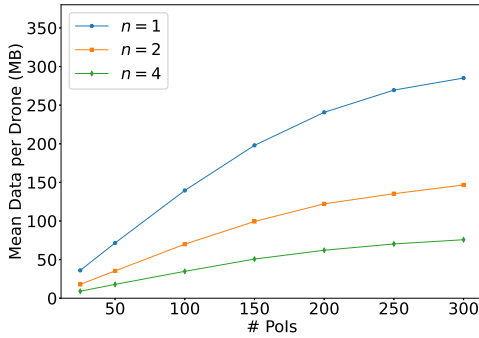


(a)

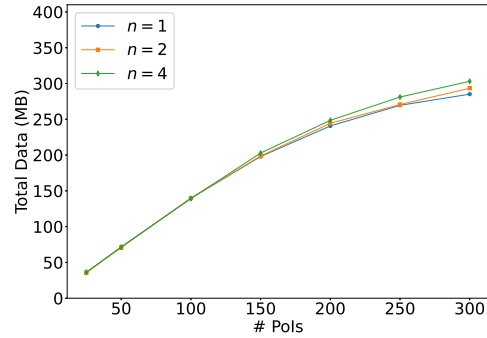


(b)

Figure D.14: Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 25m$ and spiral coverage path. The figures depict metric relative to the data collected by the drones during the mission. The data is measured in MegaBytes (MB): **(a)** Mean data per drone for missions with variable number of PoIs. **(b)** Total data collected by the swarm.



(a)



(b)

Figure D.15: Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 50m$ and parallel lines coverage path. The figures depict metric relative to the data collected by the drones during the mission. The data is measured in MegaBytes (MB): **(a)** Mean data per drone for missions with variable number of PoIs. **(b)** Total data collected by the swarm.

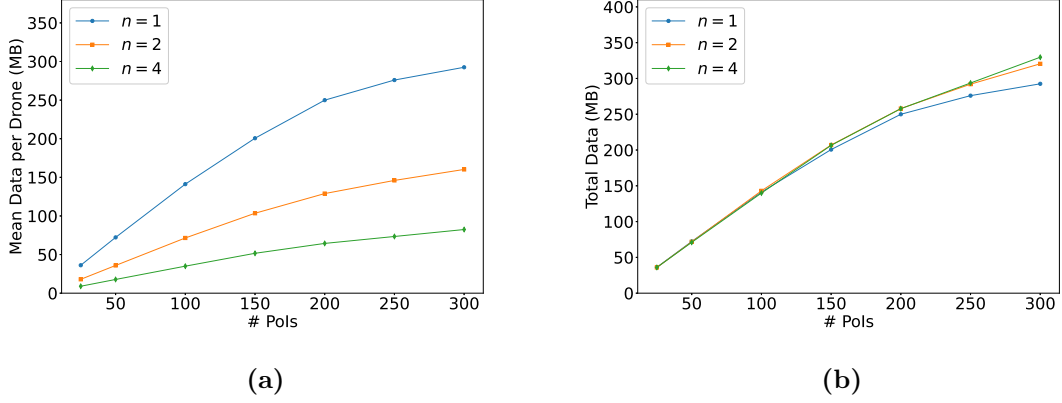


Figure D.16: Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 50$ and spiral coverage path. The figures depict metric relative to the data collected by the drones during the mission. The data is measured in MegaBytes (MB): **(a)** Mean data per drone for missions with variable number of PoIs. **(b)** Total data collected by the swarm.

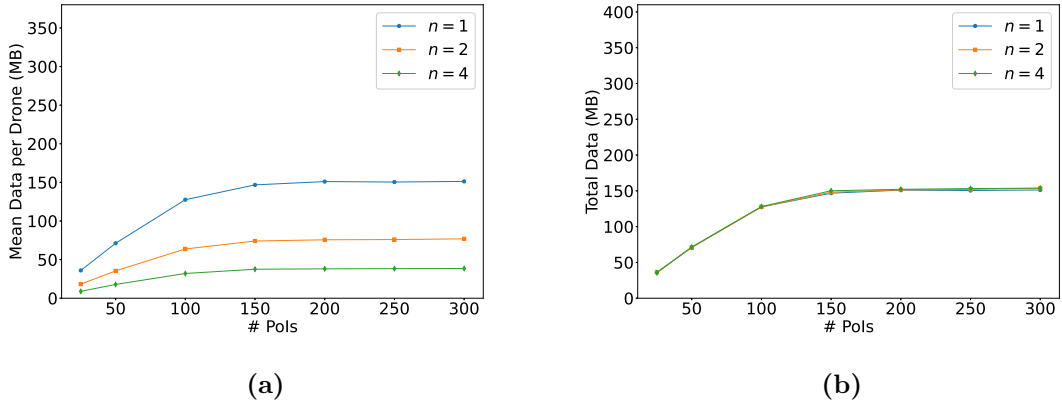


Figure D.17: Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 100m$ and parallel lines coverage path. The figures depict metric relative to the data collected by the drones during the mission. The data is measured in MegaBytes (MB): **(a)** Mean data per drone for missions with variable number of PoIs. **(b)** Total data collected by the swarm.

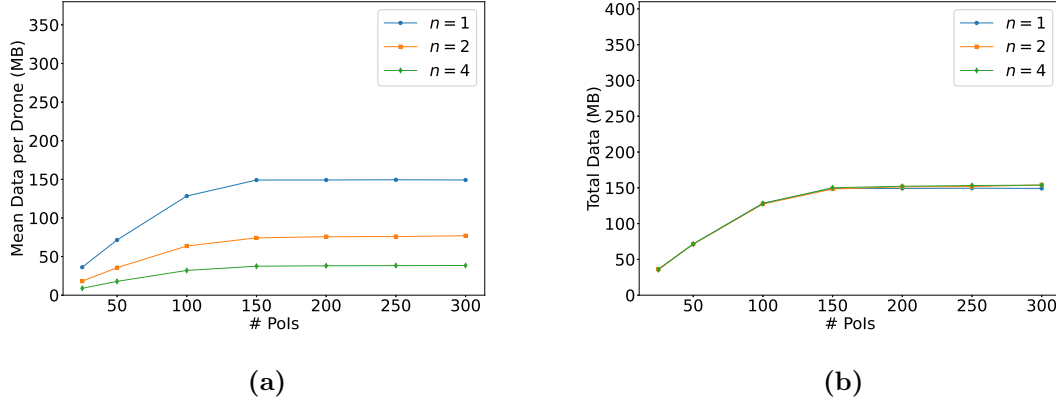


Figure D.18: Simulated results for a swarm with 1, 2 and 4 drones with $r_s = 100m$ and spiral coverage path. The figures depict metric relative to the data collected by the drones during the mission. The data is measured in MegaBytes (MB): **(a)** Mean data per drone for missions with variable number of PoIs. **(b)** Total data collected by the swarm.

D.3 Results Summary

The results collected during the simulations provide insight into the algorithm's performance. The area covered by the drones is a polygon with four (4) edges and its size remains unchanged throughout the simulations. When the drones enter the area, they choose initial and final points based on the closest vertex or border point. The swarm alignment process ensures that any part of the area will only be covered once since all drones calculate their position based on their distance from neighboring drones. At least one drone adheres to the polygon borders and the rest of the swarm simply retains a stable distance from neighboring drones.

The area coverage path is a key factor that affects the swarm's performance. Two coverage paths are examined in the simulations, parallel lines, and spiral path planning. Both can completely cover the area. Based on the simulated results, the spiral shape requires additional time and travel distance from the swarm to cover an area of the same size compared to parallel lines. This occurs since the area is covered in small portions at a time, in a linear line which equals the size

of a polygon's edge. Each time a linear coverage is complete, the covered area is removed from the polygon, hence, a new polygon is created with two smaller edges and two equal to the original size.

The parallel lines coverage path reduces the size of two polygon's edges while the other two remain at the original size for every linear coverage. The spiral path though changes the size of different edges for every linear coverage since every new formation is perpendicular to the previous. Thus, area coverage is affected by the size of the edge chosen each time by the swarm. By choosing larger sides, the final coverage performance can significantly increase. Smaller polygon sides create the need for an increased number of formation switches to reenter the area and continue the coverage.

The drone's scanning radius can affect the swarm's ability to locate and collect information from points of interest. The simulated results depict variable percentages of covered points of interest. A drone, based on the algorithm's functionality, can collect information from one point of interest at a time. It picks the one closest to it. The larger the number of points in the area, the harder it is to locate and successfully collect the information from a point of interest. An area with the same number of PoIs which also have the same locations and information to be collected from can have variable performance based on the swarm's speed and moving direction. The algorithm states that speed is prone to deviation based on the required information from a located point of interest. Moving direction is adjusted by the area coverage path.

Edge cases, where a PoI's distance within the scanning radius is very small, can significantly reduce the algorithm's effectiveness. Time and mean speed of the swarm are affected, hence the swarm remains in an area for much longer compared to normal flying speed. The entire swarm reduces its speed to the currently broadcast lower speed. This is a condition stated by the algorithm and all drones have to comply with it.

If one drone requires more time, the others have to follow. This results in two cases. First, the other drones also have more time for the currently covered area which means that if they have located lots of PoIs, they have the opportunity to collect a larger amount of information since the normal speed is occasionally a deterrent. In this case, although time has significantly increased, the swarm benefits since it has collected information that could have been lost otherwise. In the second

case, the other drones in the swarm do not have any points of interest within their scanning radius. This creates an overhead to the swarm for which no obvious benefit is observed.

Chapter E

Conclusions and Future Work

ROBOTIC systems have gained significant ground over the last decades due to the technological advances towards the miniaturization of computing devices. Battery technology, sensors, and material science are often combined and create robotic systems which can tackle complex tasks. Human intervention has shifted from full control of robotic devices to a more pathetic role. Drones or unmanned areal vehicles are a great example of such robotic systems. The first appearance of drones dates back to 1903 when the US military developed the first plane which required no crew members to fly and aimed at battlefield applications.

Human operators were an integral part of drone operation since the technology at the time, only allowed the wireless transmission of simple commands which helped the drone navigate the environment and take action on the battlefield. Ever since computer systems have advanced to the point where drones can navigate a natural environment using information collected during their flight. Sensors such as proximity calculators, accelerometers, gyroscopes, ultrasonic, LiDAR, GPS, and cameras can effectively provide a flying drone with enough information to help it locate and avoid obstacles, recognize predefined targets and fly towards geographical locations.

Environmental information collection is an important step, but accurate processing and analysis are also key to achieving certain tasks. Modern applications which use drones are not limited to warfare. Drones are being employed to monitor forests for potential fires, for crisis response, for accurate area mapping, for oil spill detection and monitoring, and many others. Such applications usually employ multiple drones which are programmed to operate under an algorithm that coordinates the

drones towards a swarm behavior.

Many modern applications are developed based on the principle of autonomous drone navigation and swarm intelligence. The development of algorithms that take advantage of information measured locally by the drone's sensors can potentially remove human error in operations since external input to the system is minimized. The drones collect information from the surrounding environment and take action based on the swarm's state. Communications between the swarm's drones are employed to increase awareness and provide more reliable results for cooperative task completion.

Drone swarms and swarm intelligence are especially important for applications where large areas need to be monitored. For example, early detection of forest fire is considered a mission in which drones are required to cover large, and hard-to-reach areas where human navigation is either difficult –using ground vehicles– or not cost-effective –using areal vehicles–. Drones, on the other hand, despite having limited energy resources can cooperate and achieve an area coverage while operating within the available resource limits of a single drone. The area can be covered by the swarm when each drone flies over smaller segments and collects information that, combined, can effectively provide knowledge about the entire area.

The field which studies the coverage problem is called Coverage Path Planning (CPP). Over the years many algorithmic processes have been used to tackle the CPP problem. Swarm intelligence algorithms inspired by nature, such as ant and bee colony intelligence. Also, heuristic solutions to problems such as the Traveling Salesman or the Vehicular Routing are employed to solve the CPP problem. Solutions to the problem are following two strategies, cellular decomposition, and no decomposition. Cellular decomposition is the process of separating the area of interest into smaller segments or cells which simplify the area coverage by creating simply structured polygons. No decomposition is used for areas where obstacles do not exist and the area of interest has a simple polygon defining it.

In the current thesis, no decomposition techniques are employed and the coverage of simple polygons is studied for two different paths, parallel lines –or back-and-forth movement– and spiral path. An algorithm is developed to tackle the problem of CPP which uses distributed operations with very basic knowledge and creates paths for each drone in the swarm. The algorithm coordinates the drones towards the full area coverage which is achieved. The drones are also able to locate and collect

information from points of interest.

E.1 Conclusions

A simulated environment is developed to examine the algorithm's effectiveness. The environment's variables are, the number of drones in the swarm, the number of points of interest in the area, the scanning radius of each drone, and the coverage path. Multiple executions with similar variable values and different points of interest locations are executed to determine the algorithm's ability to locate points of interest and collect information from them.

The results depict that parallel lines coverage is more effective for the same area size in square polygons compared to spiral coverage. This is attributed to the fact that the drones cover the area until they reach the borders. Each time this happens, they create a new area based on the original which does not include the already covered part. As a result, the drones are outside the new area of interest and they have to switch their formation to reenter and continue with the coverage.

The spiral path creates smaller edges since a linear area with a size equal to the scanning radius multiplied by the edge size is removed. The parallel path leads the drones towards covering an area of the same edge size each time. In the latter, the swarm has to switch its formation fewer times compared to the latter, hence the drones have to travel less distance to cover the entire area. Coverage time is also affected by this behavior.

The swarm is better at detecting and collecting information from points of interest when the scanning radius has its lowest value, 25 meters. For the spiral coverage, the number of times the swarm has to slow down for a point of interest has the highest values. The number of points of interest fully discovered is about the same compared to the parallel coverage but to achieve it, the swarm has to decrease its speed 50 more times on average.

E.2 Future Work

The algorithm provides the swarm with the necessary steps for the full area coverage and the collection of as much information as possible from points of interest, but

there is room for improvements. First, when the drones locate a point of interest, it calculates the amount of time this particular point will remain within its radius while it flies over it.

All further actions including the speed adjustment, are taken based on this single point of interest. The drone then focuses all its attention on collecting the information from this point ignoring other points within the scanning radius. A possible improvement could help the drones miss fewer points of interest by choosing the point that will remain within the radius for less time compared to the rest of the points. Also, speed can be adjusted based on all located points and the amount of information required from each one.

The current algorithm states that a drone collects information from a single point of interest at a time. Applications that include detection of possible points of interest using cameras can have multiple-point detection. This limitation is proposed for low-cost systems where multiple antennas or advanced ones which can communicate with multiple wireless devices are not an option. A possible improvement may include multi-point information collection which will increase the number of detected points.

Finally, the algorithm provides path planning that does not take into account obstacles. The path planning can be improved by providing real-time obstacle avoidance which will require the swarm's formation to alter in real-time and find a path around the obstacle.

Bibliography

- [1] D. Hambling, *Swarm Troopers: How small drones will conquer the World*. Publishing services provided by Archangel Ink, 2015.
- [2] F. Yamazaki and W. Liu, “Remote sensing technologies for post-earthquake damage assessment: A case study on the 2016 kumamoto earthquake,” in *6th Asia Conference on Earthquake Engineering*, p. 8p, 2016.
- [3] D. Kinaneva, G. Hristov, J. Raychev, and P. Zahariev, “Early forest fire detection using drones and artificial intelligence,” in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1060–1065, IEEE, 2019.
- [4] H. X. Pham, H. M. La, D. Feil-Seifer, and M. Deans, “A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6648–6653, IEEE, 2017.
- [5] I. Hong, M. Kuby, and A. Murray, “A deviation flow refueling location model for continuous space: a commercial drone delivery system for urban areas,” in *Advances in Geocomputation*, pp. 125–132, Springer, 2017.
- [6] D. Bamburly, “Drones: Designed for product delivery,” *Design Management Review*, vol. 26, no. 1, pp. 40–48, 2015.
- [7] J. A. Guerrero and Y. Bestaoui, “Uav path planning for structure inspection in windy environments,” *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1, pp. 297–311, 2013.

- [8] W. Chang, G. Yang, J. Yu, Z. Liang, L. Cheng, and C. Zhou, “Development of a power line inspection robot with hybrid operation modes,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 973–978, IEEE, 2017.
- [9] M. Champion, P. Ranganathan, and S. Faruque, “Uav swarm communication and control architectures: a review,” *Journal of Unmanned Vehicle Systems*, vol. 7, no. 2, pp. 93–106, 2018.
- [10] U. R. Mogili and B. Deepak, “Review on application of drone systems in precision agriculture,” *Procedia computer science*, vol. 133, pp. 502–509, 2018.
- [11] V. Lomonaco, A. Trotta, M. Ziosi, J. D. D. Y. Avila, and N. Díaz-Rodríguez, “Intelligent drone swarm for search and rescue operations at sea,” *arXiv preprint arXiv:1811.05291*, 2018.
- [12] F. Aznar, M. Sempere, M. Pujol, R. Rizo, and M. Pujol, “Modelling oil-spill detection with swarm drones,” in *Abstract and Applied Analysis*, vol. 2014, Hindawi, 2014.
- [13] P. Odonkor, Z. Ball, and S. Chowdhury, “Distributed operation of collaborating unmanned aerial vehicles for time-sensitive oil spill mapping,” *Swarm and Evolutionary Computation*, vol. 46, pp. 52–68, 2019.
- [14] M. S. Innocente and P. Grasso, “Self-organising swarms of firefighting drones: Harnessing the power of collective intelligence in decentralised multi-robot systems,” *Journal of Computational Science*, vol. 34, pp. 80–101, 2019.
- [15] E. Ausonio, P. Bagnerini, and M. Ghio, “Drone swarms in fire suppression activities: A conceptual framework,” *Drones*, vol. 5, no. 1, p. 17, 2021.
- [16] C. Yuan, Y. Zhang, and Z. Liu, “A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques,” *Canadian journal of forest research*, vol. 45, no. 7, pp. 783–792, 2015.
- [17] A. Tahir, J. Böling, M.-H. Haghbayan, H. T. Toivonen, and J. Plosila, “Swarms of unmanned aerial vehicles—a survey,” *Journal of Industrial Information Integration*, vol. 16, p. 100106, 2019.

- [18] H. Chen, X.-m. Wang, and Y. Li, “A survey of autonomous control for uav,” in *2009 International Conference on Artificial Intelligence and Computational Intelligence*, vol. 2, pp. 267–271, IEEE, 2009.
- [19] C. T. Cicek, H. Gultekin, B. Tavli, and H. Yanikomeroglu, “Uav base station location optimization for next generation wireless networks: Overview and future research directions,” in *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*, pp. 1–6, IEEE, 2019.
- [20] P. Katsigiannis, L. Misopolinos, V. Liakopoulos, T. K. Alexandridis, and G. Zalidis, “An autonomous multi-sensor uav system for reduced-input precision agriculture applications,” in *2016 24th Mediterranean Conference on Control and Automation (MED)*, pp. 60–64, IEEE, 2016.
- [21] B. Zhao, X. Chen, X. Zhao, J. Jiang, and J. Wei, “Real-time uav autonomous localization based on smartphone sensors,” *Sensors*, vol. 18, no. 12, p. 4161, 2018.
- [22] P. Doherty and P. Rudol, “A uav search and rescue scenario with human body detection and geolocalization,” in *Australasian Joint Conference on Artificial Intelligence*, pp. 1–13, Springer, 2007.
- [23] A. Gaszczak, T. P. Breckon, and J. Han, “Real-time people and vehicle detection from uav imagery,” in *Intelligent Robots and Computer Vision XXVIII: Algorithms and Techniques*, vol. 7878, p. 78780B, International Society for Optics and Photonics, 2011.
- [24] A. L. Alfeo, M. G. Cimino, N. De Francesco, M. Lega, and G. Vaglini, “Design and simulation of the emergent behavior of small drones swarming for distributed target localization,” *Journal of computational science*, vol. 29, pp. 19–33, 2018.
- [25] N. Jin, R. Ma, Y. Lv, X. Lou, and Q. Wei, “A novel design of water environment monitoring system based on wsn,” in *2010 International conference on computer design and applications*, vol. 2, pp. V2–593, IEEE, 2010.
- [26] J. Carle and D. Simplot-Ryl, “Energy-efficient area monitoring for sensor networks,” *Computer*, vol. 37, no. 2, pp. 40–46, 2004.

- [27] A. Ahmed, J. Ali, A. Raza, and G. Abbas, “Wired vs wireless deployment support for wireless sensor networks,” in *TENCON 2006-2006 IEEE Region 10 Conference*, pp. 1–3, IEEE, 2006.
- [28] N. A. B. Ab Aziz, A. W. Mohemmed, and M. Y. Alias, “A wireless sensor network coverage optimization algorithm based on particle swarm optimization and voronoi diagram,” in *2009 international conference on networking, sensing and control*, pp. 602–607, IEEE, 2009.
- [29] J. Ren, Y. Zhang, K. Zhang, A. Liu, J. Chen, and X. S. Shen, “Lifetime and energy hole evolution analysis in data-gathering wireless sensor networks,” *IEEE transactions on industrial informatics*, vol. 12, no. 2, pp. 788–800, 2015.
- [30] C. Yuan, Z. Liu, and Y. Zhang, “Fire detection using infrared images for uav-based forest fire surveillance,” in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 567–572, IEEE, 2017.
- [31] D. Popescu, C. Dragana, F. Stoican, L. Ichim, and G. Stamatescu, “A collaborative uav-wsn network for monitoring large areas,” *Sensors*, vol. 18, no. 12, p. 4202, 2018.
- [32] A. Bürkle, F. Segor, and M. Kollmann, “Towards autonomous micro uav swarms,” *Journal of intelligent & robotic systems*, vol. 61, no. 1, pp. 339–353, 2011.
- [33] D. Câmara, “Cavalry to the rescue: Drones fleet to help rescuers operations over disasters scenarios,” in *2014 IEEE Conference on Antenna Measurements & Applications (CAMA)*, pp. 1–4, IEEE, 2014.
- [34] D. Kingston, R. W. Beard, and R. S. Holt, “Decentralized perimeter surveillance using a team of uavs,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1394–1404, 2008.
- [35] F. Yang, P. Wang, Y. Zhang, L. Zheng, and J. Lu, “Survey of swarm intelligence optimization algorithms,” in *2017 IEEE International Conference on Unmanned Systems (ICUS)*, pp. 544–549, IEEE, 2017.

- [36] O. Cheikhrouhou and I. Khoufi, “A comprehensive survey on the multiple traveling salesman problem: Applications, approaches and taxonomy,” *Computer Science Review*, vol. 40, p. 100369, 2021.
- [37] A. T. Khader, M. A. Al-betar, and A. A. Mohammed, “Artificial bee colony algorithm, its variants and applications: a survey,” 2013.
- [38] D. Teodorović, “Bee colony optimization (bco),” in *Innovations in swarm intelligence*, pp. 39–60, Springer, 2009.
- [39] D. Karaboga and B. Akay, “A survey: algorithms simulating bee swarm intelligence,” *Artificial intelligence review*, vol. 31, no. 1-4, p. 61, 2009.
- [40] M. Gambardella, M. B. A. Martinoli, and R. P. T. Stützle, “Ant colony optimization and swarm intelligence,” in *5th international workshop*, Springer, Springer, 2006.
- [41] V. Selvi and R. Umarani, “Comparative analysis of ant colony and particle swarm optimization techniques,” *International Journal of Computer Applications*, vol. 5, no. 4, pp. 1–6, 2010.
- [42] Q.-w. Yang, S.-n. Ruan, J.-f. Chen, and B. LI, “Application survey on swarm intelligence in the traveling salesman problem [j],” *Techniques of Automation & Applications*, vol. 8, pp. 1–6, 2016.
- [43] M. Reed, A. Yiannakou, and R. Evering, “An ant colony algorithm for the multi-compartment vehicle routing problem,” *Applied Soft Computing*, vol. 15, pp. 169–176, 2014.
- [44] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, “Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey,” *Networks*, vol. 72, no. 4, pp. 411–458, 2018.
- [45] T. M. Cabreira, L. B. Brisolara, and P. R. Ferreira Jr, “Survey on coverage path planning with unmanned aerial vehicles,” *Drones*, vol. 3, no. 1, p. 4, 2019.
- [46] L. Nam, L. Huang, X. J. Li, and J. Xu, “An approach for coverage path planning for uavs,” in *2016 IEEE 14th international workshop on advanced motion control (AMC)*, pp. 411–416, IEEE, 2016.

- [47] Y. Li, H. Chen, M. J. Er, and X. Wang, “Coverage path planning for uavs based on enhanced exact cellular decomposition method,” *Mechatronics*, vol. 21, no. 5, pp. 876–885, 2011.
- [48] K. McGuire, C. De Wagter, K. Tuyls, H. Kappen, and G. C. de Croon, “Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment,” *Science Robotics*, vol. 4, no. 35, 2019.
- [49] Y. Xia, C. Chen, J. Shi, Y. Liu, and G. Li, “Two-layer path planning for multi-area coverage by a cooperated ground vehicle and drone system,” 2020.
- [50] K. Skiadopoulos, K. Giannakis, A. Tsipis, K. Oikonomou, and I. Stavrakakis, “Impact of drone route geometry on information collection in wireless sensor networks,” *Ad Hoc Networks*, vol. 106, p. 102220, 2020.
- [51] H. Choset, “Coverage for robotics—a survey of recent results,” *Annals of mathematics and artificial intelligence*, vol. 31, no. 1, pp. 113–126, 2001.

Appendices

Appendix I

Notations

r_s	The drone's scanning radius.
m_d	The drone's movement direction.
p_i^a	The initial point found after i attempts.
p_i^c	The final point found after i attempts.
d_s	The distance for which a PoI will remain within a drone's scanning radius.
b_c	The closest polygon border point.

Appendix II

Abbreviations

CPP	Coverage Path Planning
UAV	Unmanned Aerial Vehicle
PoI	Point of Interest
PSO	Particle Swarm Optimization
VTOL	Vertical Take-Off and Landing
TSP	Travelling Salesman Problem
VRP	Vehicular Routing Problem
BCO	Bee Colony Optimization

Appendix III

Source Code

```
void UAVNode::handleMessage(cMessage *msg) {
if (discoveryState == CONNECTED) {
    // While collecting data from a PoI,
    cDisplayString& displayString =
        currentlyConnectedNode.hiddenNode->getDisplayString();
    const char * image = displayString.getTagArg("i", 0);
    string image_string = string(image);
    // the drone changes the PoI's image to optically
    // detect it in the user interface
    if (image_string == "uav-swarm/sensor_wireless_s") {
        displayString.setTagArg("i", 0,
                                "uav-swarm/sensor_s");
    } else {
        displayString.setTagArg("i", 0,
                                "uav-swarm/sensor_wireless_s");
    }
    // Finally it updates the collected data and
    // stops the collection when it reaches 0
    updateDataRemainingForCollection();
}
updateDistancesWithHidden(); // Updating the distances with th PoIs
if (msg->isSelfMessage()) {
    if (state == BORDER_ALIGN) {
        borderAlign();
    } else if (state == SCANNING) {
        scanning();
    }
}
```

```

} else { // Handling remote messages
    MessageType type = MessageType(msg->getKind());
    if (type == CLOSEST_FROM_POLYGON_POINT) {
        if (state == SCANNING) {
            // Keeping the closest border point
            // message in memory until
            // the final point is reached.
            // Then all messages are handled
            closestPointMessages.push_back(msg);
        } else if (state == BORDER_ALIGN
            || state == WAITING_FOR_ALIGNMENT) {
            // If the drone is also waiting for
            // alignment it will
            // handle the message immediately
            handleClosestPointMessage(msg);
        }
    } else if (type == READY_TO_SCAN) {
        // Message that informs the swarm
        // that the initial point has been
        // reached and the drone that sent
        // the message is hovering over it
        // and waiting for the rest of the
        // swarm to initialize the area coverage
        handleReadyToScan(msg);
    } else if (type == SPEED_CHANGE) {
        // This message is transmitted
        // when a drone requires more
        // time for data collection from a PoI
        handleNewSpeed(msg);
    }
}
}

void UAVNode::checkArea(double area) {
    if (area > 0) { // Keep on covering if there is still area left
        // Time to switch formation and find new
        // initial and final points
        state = BORDER_ALIGN;
        // Disconnecting from any PoIs since the drone is not scanning
        disconnectFromHiddenIfLostProximity();
        // Changing direction only for parallel coverage
    }
}

```

```

    if (parallel_scanning) {
        globalDirection = Direction(-1 * int(globalDirection));
    }
    if (numberOfUAVs == 1) { // Code for a single drone
        PointInfo closestPoint = getClosestPoint(0);
        pair<Point, Point> initialAndFinalPoints =
            getInitialAndFinalPoint(scanningRadius,
                                    closestPoint, true);
        // New target, the initial point
        target = initialAndFinalPoints.first;
        finalPoint = initialAndFinalPoints.second;
        // Maintaining the array indexes for the border points
        // that will require updating later
        borderPoints[initialToAdd.index_p] = initialToAdd.point;
        borderPoints[finalToAdd.index_p] = finalToAdd.point;
        scheduleAt(simTime(), new cMessage());
        return;
    }
    // Sending the current location and the closest border point
    broadcastClosestPoint(closestPoint.index_p,
                        closestPoint.distance);
    // Handling closest point messages while waiting for alignment
    if (closestPointMessages.size() > 0) {
        for (int i = 0; i < closestPointMessages.size(); i++) {
            handleClosestPointMessage(closestPointMessages[i]);
        }
        closestPointMessages.clear();
    }
}

void UAVNode::scanning() {
    // Removing the distance traveled until
    // the first border is reached.
    // Calculate the distance and time for the area coverage
    if (isFirstAlignment) {
        timeToRemove = simTime();
        distanceToRemove = distanceTravelled;
        isFirstAlignment = false;
    }
    // Moving parallel to the border edge until

```

```

        // the final point is reached
        moveToNextParallelPoint();
        // Connecting with PoI in proximity
        connectWithHiddenInProximity();
        // disconnecting from PoI when lost proximity
        disconnectFromHiddenIfLostProximity();

        // While final point not reached
        if (loc.distanceTo(finalPoint) > 4) {
            lastStep = simTime();
            scheduleAt(simTime() + timeStep, msg);
        } else {
            loc = finalPoint; // Correcting inaccuracies in the graphics
            if (isCanvasManager) { // Updating the canvas border points
                EV << "Changed_Canvas_Points!" << endl;
                mist_polygon->setPoint(initialToAdd.index_p,
                                       initialToAdd.point);
                mist_polygon->setPoint(finalToAdd.index_p,
                                       finalToAdd.point);
                isCanvasManager = (numberOfUAVs == 1) ? true : false;
                borderPoints[initialToAdd.index_p] = initialToAdd.point;
                borderPoints[finalToAdd.index_p] = finalToAdd.point;
            }
            // Finding the closest border point which will be used
            // for the new initial and final points
            PointInfo closestPoint = getClosestPoint(0);
            double area = getAreaSize(); // Remaining area for coverage
            checkArea(area);
        }
    }

void UAVNode::borderAlign(){
    // Speed equals the switched speed which might be the
    // original or the decreased speed
    speed = switchedSpeed;
    // Calculating the distance that will be traveled for the
    // next second
    double distance = speed * timeStep.dbl();
    // The new x and y values on the canvas
    loc.x += distance * cos(heading);

```

```

loc.y += distance * sin(heading);
// Adding the current distance in the total travel distance
distanceTravelled += distance;
refresh(); // Refresing the display
// Calculating the vector from the current location to the
// target location
Point vectorToTarget = target - loc;
double targetDirection = atan2(vectorToTarget.y, vectorToTarget.x);
// Getting the heading direction used in the next location switch
heading = targetDirection;
lastStep = simTime(); // Keeping the event time to use it later
// Checking if distance from initial point is small enough in
// order to switch state and start scanning.
Point distanceFromTargetVector = loc - target;
if (loc.distanceTo(target) <= 4) {
    switchFromBorderAlign(); // Switching to scanning
} else {
    // Scheduling next event until the borders are reached
    scheduleAt(simTime() + timeStep, msg);
}
// Disconnecting from any PoI while in border align
disconnectFromHiddenIfLostProximity();
}

```

Index

- alignment, 15
- Area borders, 16
- area coverage, 3
- area monitoring, 14
- area of interest, 3
- Autonomous, 2
- base station, 3
- border point, 16
- cells, 10
- cellular decomposition, 11
- collect information, 20
- cover, 10
- coverage, 2
- coverage path, 10
- Coverage Path Planning, 7
- coverage time, 26
- CPP, 7
- drone, 1
- final point, 17
- formation, 12
- human intervention, 14
- initial points, 17
- locate PoIs, 20
- location, 14
- Movement direction, 16
- navigation, 4
- neighboring drones, 14
- no decomposition, 12
- parallel, 11
- perpendicular, 29
- Point of interest, 4
- scanning radius, 10
- scouting, 14
- self-organization, 15
- speed, 14
- spiral, 12
- swarm, 2
- travel distance, 12
- UAV, 5
- Unmanned Aerial Vehicle, 5