

UAV Coverage Path Planning for Autonomous Deer Detection in the Wild Driven by Ant Colony Optimization

Mario Jerez ¹, Khoi Duong ¹, Jack Swanberg ¹, Horoom Bula ¹, Ebas Temesgen ¹, and Maria Gini ¹

Abstract—Unmanned aerial vehicle (UAV) technology can reduce the damage caused by deer on farmlands through autonomous detection and deterrence. Coverage path planning algorithms used in the literature mostly use predictable searching techniques that trace a path across an area in a back-and-forth pattern. This paper proposes a coverage path planning algorithm that utilizes Ant Colony Optimization (ACO) to generate energy-efficient and less predictable search strategies for detecting deer. By comparing the energy efficiency of our approach with the commonly used back-and-forth algorithm, we find that ACO generates more energy-efficient solutions to the coverage path planning problem for a local farm. Based on this finding, we recommend that our approach be used to generate flight paths for UAVs in fully implemented wild animal detection and deterrence systems. A more energy-efficient search algorithm gives the UAV more time to search for and deter deer.

I. INTRODUCTION

Using unmanned aerial vehicles (UAVs) to detect and deter wild animals has gained popularity due to their ability to quickly survey an area and scare away animals by flying towards them [1]–[3]. In this paper, we propose the use of Ant Colony Optimization (ACO) [4] to improve the fuel efficiency of UAV coverage path planning algorithms when searching for wild animals such as deer. We demonstrate that with our proposed methodology, one or multiple UAVs can fully cover an area using an energy efficient flight path, extending the area that a single drone can cover. We also propose that our coverage algorithm may generate flight paths that are harder for deer to predict and evade detection.

Our research is motivated by the damage that deer cause to farmers' crops during the growing season. We reached out to farmers in Minnesota by posting to a sustainable agriculture listserv, an email subscription system, wanting to learn about the problems they face. From the farmers who shared their experiences, the most common concerns they mentioned had to do with wild animals such as deer, voles, and gophers damaging their crops. Deer damage in particular has caused disagreements among farmers, hunters, and the Department of Natural Resources over how the deer population should be controlled [5]. We hope that this research will help alleviate the challenges farmers face and resolve issues caused by deer on farms.

*Work supported in part by the AI-CLIMATE Institute
All authors are in the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, USA.
{jerez005, duong245, swanb045, bula0023, temes021, gini}@umn.edu)

A. Contributions

The key contributions of this paper are:

- 1) A method for detecting deer on farms that can be incorporated into a deterrence system for farmers.
- 2) A coverage path planning algorithm applicable for one or more unmanned aerial vehicles (UAVs) that maximizes energy efficiency by using Ant Colony Optimization (ACO) with built-in obstacle avoidance.

This paper is organized as follows. Section II reviews related work in detecting and deterring wild animals from private property and other prior work related to our approach to wild deer deterrence. Section III explains how the different components of our solution work together to address the deer problem that farmers face. Section IV describes our proposed use of the Ant Colony Algorithm (ACO) to find optimal solutions to the path coverage planning problem. Section V outlines our experimental setup and how our algorithm is applied to a real-life scenario. Section VI presents our experimental results. Section VII concludes the paper, discussing our findings and interpretation of results, and outlines avenues for future research.

II. RELATED WORK

A. Deer detection and deterrence

Our research contributes to the body of work done in the autonomous detection and deterrence of large animals, such as deer. While substantial work has been done in autonomous deterrence of small pests such as birds [2], [3], [6], the autonomous detection and deterrence of larger animals has been less thoroughly explored [7]. Important prerequisite technologies for deer detection have been developed, such as stationary cameras that detect deer using computer vision techniques [8], [9] and even UAVs that autonomously detect deer [10]. Approaches used to detect and deter birds, such as using stationary cameras for detection, estimating their coordinates, and deploying UAVs in the area for deterrence [2], [6] could be applied to deer. However, for farmers with medium-to-large farms, setting stationary cameras throughout the farm to surveil its entirety may be infeasible due to installation costs and the challenge of supplying the cameras with constant power.

Instead, our work builds on approaches to detecting wild animals on agricultural land using UAVs, without the aid of ground stationary cameras. Our contribution more aligns with existing work in deer detection that plans a coverage path through waypoints virtually embedded throughout a field for a UAV to follow [11]. A waypoint is a node that represents

a point in physical or simulated space. Similarly to popular coverage path planning techniques [12]–[16], work in this area creates paths using a back-and-forth pattern discussed in the next subsection that steadily moves across an area. This may not be the ideal flight path to use for deer detection and deterrence due to its repetitive and predictable nature. As we demonstrate in this paper, it is also not the most energy-efficient.

B. Coverage path planning algorithms

A commonly used baseline for area coverage is the *back-and-forth* coverage heuristic. The vehicle traverses the area in a series of parallel lines, turning 90° at the ends. This creates a Boustrphedon pattern. Due to its simplicity and effectiveness in simple spaces, it has been extensively researched and used.

To handle more complex environments, researchers have proposed enhancements that build on this basic pattern. Cellular decomposition methods, for example, divide the environment into smaller subregions where the back-and-forth strategy can be applied independently [17]–[19]. More recent work has introduced additional heuristics to adapt the algorithm complex spaces. Mu et al. [14] propose a continuity constraint to prevent the agent from splitting the space prematurely, a greedy strategy for turn direction (instead of always turning back), and an A*-based module to navigate out of confined areas. Cao et al. [20] introduces the Improved Probabilistic Roadmap, which first generates a set of parallel “straight paths”, then connects them using a Traveling Salesperson Problem (TSP) [21] heuristic to ensure full coverage while minimizing transition costs.

Despite its widespread use, the back-and-forth strategy has notable drawbacks in dynamic and adversarial settings. One key concern is predictability: all algorithms based on the back-and-forth heuristic produce a characteristic pattern of 90° turns and repeated long parallel paths. In deer deterrence contexts, such predictability may lead to habituation [22], [23] and risk deer learning the drone’s pattern and adapting their behavior to avoid being detected.

Other research in deterring pest birds from agricultural crops accounts for habituation by using an occupancy grid map algorithm with multiple UAVs [3]. We attempt to prevent habituation by using a coverage path planning algorithm that does not always produce the same path and that generates less predictable flight patterns. We also use multiple UAVs to search an area; but, instead of collaboratively searching the same space, we partition the area into subsections small enough to be covered by a single UAV.

Several techniques have been used to minimize energy consumption in coverage path planning algorithms for UAVs. Prior work has improved energy efficiency by setting the height of the UAV to the maximum altitude that meets resolution criteria, strategically placing waypoints, and creating energy aware back-and-forth paths through them [24]. Other work has solved the energy efficient coverage path planning problem as a TSP to generate a flight path [25]. Another approach has been to divide a space into sub-areas,

where the path between sub-areas is solved as a TSP with ACO, and each subarea gets covered with a back-and-forth algorithm [26]. A more recent approach has been to solve the coverage path planning problem as a TSP by using ACO to send a UAV to all regions while minimizing the total distance covered [27]; however, the distance that a UAV travels is not the only indicator of its energy consumption since total rotation is also a primary contributor to a UAV’s energy expenditure [28]. This paper combines aspects of these prior work by using ACO to find paths that hit all waypoints in a region while minimizing energy cost by taking into account total distance and rotation. In order to maximize energy efficiency and thus area coverage capabilities, our proposed algorithm solves the Energy Efficient Coverage Path Planning (EECPP) problem, in which a UAV must fly through all waypoints exactly once and return to the starting waypoint while minimizing distance traveled and degrees rotated.

C. Ant Colony Optimization

Ant Colony Optimization (ACO) is a method used for solving combinatorial optimization problems that was inspired by ants’ abilities to collaboratively find the shortest path between their nest and a food source [29], [30]. Similarly to real ants, virtual ants in ACO communicate with one another indirectly by depositing and sensing pheromones in the solution space. In ACO, ants find optimal solutions by making decisions based on the heuristic values of the options available to them and the amount of pheromones placed on those options by ants from previous epochs. Both the heuristic values and pheromone levels are indicators of how favorable an option is, and ants are more likely to choose those that seem more favorable, i.e. have higher heuristic values and pheromone levels. Taking TSP as an example, an ant decides which node to visit based on its distance and the amount of pheromone that is left on the path to that waypoint. After all ants create their own solutions, pheromone values on the edges get updated such that better performing ants increase the pheromones on edges that are a part of their solution more than lower performing ants do. Over time, edges that are a part of better solutions get higher pheromone values, making ants more likely to traverse them. We use ACO because it has shown to be effective at solving myriad combinatorial optimization problems [4], including TSP [31], which is similar to the problem that we solve in this paper.

III. SYSTEM ARCHITECTURE

As illustrated in Figure 1, we begin by plotting waypoints over a 2D representation of a farm. Tall structures that a UAV could collide with (visualized as orange shapes in the figure) are included in the 2D representation. Edges (visualized as lines) are added between all possible combinations of waypoints to make a complete graph. An edge represents a potential straight-line path between two waypoints that a UAV could traverse.

In order to avoid collision with structures on the farm, edges that would cause a UAV to collide or come close to

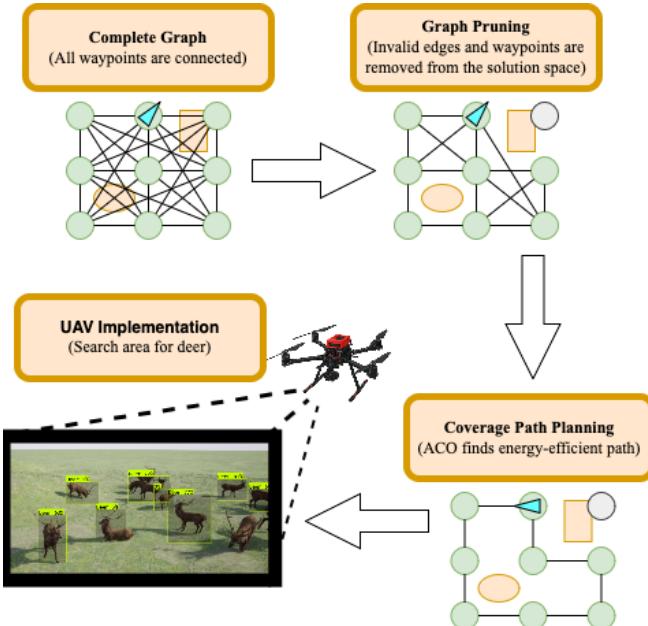


Fig. 1. Illustration of our methodology for energy-efficient deer search.

colliding with an obstacle are removed from the graph. This includes edges connected to waypoints that are too close to obstacles, which we refer to as invalid waypoints. We then use ACO to generate an energy-efficient path through all waypoints that returns to the starting waypoint since the UAV is expected to begin and end at its charging station. The 2D coordinates of the waypoints are then plotted as 3D coordinates on a farm. A UAV can then follow the path generated by ACO while using computer vision techniques to detect deer from some configurable height off of the ground.

IV. PATH COVERAGE PLANNING ALGORITHM

ACO is a promising tool for solving our coverage path planning problem in an energy-efficient way due to its high performance in solving the similar TSP [32]–[34]. We implement both Ant System (AS) [4]—the original ACO algorithm—and Max-Min Ant System (MMAS) [33], [34], which has performed particularly strongly in TSP [31].

A. Implementing the ACO Algorithm

While in AS, all ants deposit pheromones onto edges they traverse in proportion to how well they perform, in MMAS, only the best ant deposits pheromones onto its traversed edges [4], [33], [34]. This accelerates the performance of ants in future iterations as it exploits the solution of the top-performing ant. At the same time, exploration of new paths is encouraged and premature convergence on suboptimal tours is discouraged by limiting the pheromone level on any edge to be no greater than a maximum pheromone level τ_{max} and no less than a minimum pheromone level τ_{min} . Formally, $\tau_{min} \leq \tau_{ij} \leq \tau_{max}$ where τ_{ij} is the pheromone value on the undirected edge connecting node i to node j . AS has no such bounds.

Algorithm 1 Find Energy-Efficient Path with ACO

```

1: Set parameters
2: Initialize pheromone values
3: for  $t \leftarrow 1$  to maximumIterations do
4:   ConstructAntSolution
5:   UpdatePheromones
6: end for

```

1) Set-up: After setting the parameters shown in Table I, MMAS begins by initializing the pheromone values of all edges in the graph to τ_{max} so an edge is only favored over another based on its local heuristic function. Normally, when applying ACO to the traveling salesman problem, the heuristic function of the edge connecting nodes i and j is defined as $\eta_{ij} = \frac{1}{d_{ij}}$ where d_{ij} is the distance in meters between nodes i and j such that $1 \leq i \neq j \leq w$, where w is the number of waypoints or nodes. η_{ij} would thus indicate the favorability of a path, not considering the pheromone level on the path, as each ant prefers to take a shorter path.

However, in our application, distance and pheromone levels aren't the only indicators of a favorable path. Ants are also programmed to prefer the choice that requires the least amount of rotation. Therefore, we define the local heuristic function of choosing the edge that connects nodes i and j , given that the ant was most recently on node h , to be

$$\eta_{hij} = \frac{1}{\lambda d_{ij} + \gamma \theta_{hij}}$$

such that $h \neq i \neq j$. As depicted in Figure 2, θ_{hij} is the absolute value of the rotation required for a drone to go from node h to node i to node j , measured in degrees. λ is the energy cost for the drone to travel for one meter in a straight line, given in kJ/m . γ is the energy cost for the drone to rotate one degree, given in kJ/deg . The heuristic function is therefore the inverse of the cost of flying from waypoint i to waypoint j .

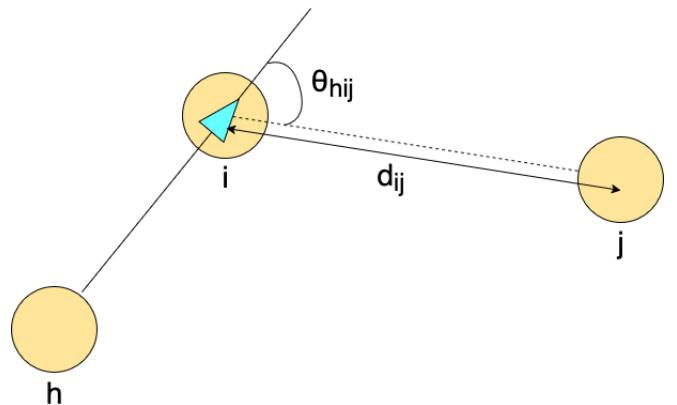


Fig. 2. Illustration of the measurements used when determining the energy cost for a drone to fly from waypoint i to waypoint j , given that it was previously on waypoint h . θ_{hij} is the degrees that the drone must rotate and d_{ij} is the linear distance it would have to travel. The drone is depicted as a blue triangle and the waypoints are orange circles.

B. Solving EECPP problem in a 2D simulated environment

Next, the optimization process begins. For every iteration in the loop in algorithm 1, all ants construct their own solutions and pheromones on edges are updated based on the performance of all ants for the AS algorithm, or only the best ant in the MMAS algorithm.

1) Construct Ant Solution: In the first iteration, each ant k , where $1 \leq k \leq n$ and n is the total number of ants, begins at node 1 and has a partial solution $s^p = \{c_{0,1}\}$, where $c_{0,1}$ is the solution component that represents traversing from node 0 to node 1. Node 0 is a fictitious node, indicating that the ant's starting node is node 1. When ant k is at node i , the probability of going to node j is given by,

$$p_{hij} = \begin{cases} \frac{\eta_{hij}^\alpha \cdot \tau_{ij}^\beta}{\sum_{c_{il} \in N(s^p)} \eta_{hil}^\alpha \cdot \tau_{il}^\beta} & \text{if } c_{ij} \in N(s^p), \\ 0 & \text{otherwise} \end{cases}$$

where $N(s^p)$ is the set of feasible components, i.e., the components c_{ij} where there exists an edge that connects nodes i and j , and j has not yet been traversed. α and β are constants that set the importance of the heuristic values and pheromone values respectively. Because an ant doesn't have a most recent node when it is on the first node, the first step in an ants solution construction warrants clarification. When an ant begins at node 1, the probability of an ant to traverse from the starting node to node j is denoted as $p_{0,1,j}$. In this case, the heuristic function $\eta_{0,1,j} = \frac{1}{\lambda d_{1j}}$ since $\theta_{0,1,j} = 0$.

2) Update Pheromones: In MMAS, after every ant k has constructed its completed solution s^k where $1 \leq k \leq n$ the best performing ant b is selected, and its solution $s^b = \{c_{01}, c_{1l_1}, c_{1l_2}, \dots, c_{lw}\}$ is used to update the pheromone value τ_{ij} of each solution component $c_{ij} \in s^b$, i.e. each edge that connects node i to node j , where $i \neq 0$. The updated pheromone value for all edges in the graph is given by,

$$\tau_{ij} \leftarrow [(1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij}^b] \tau_{\min}^{\max}$$

where ρ is the evaporation of the pheromone. In AS, edge pheromone values are updated by all ants instead of just the best ant, and the values are not bounded by τ_{\min} and τ_{\max} . $\Delta\tau_{ij}^k$ is the pheromone that an ant k adds to c_{ij} which is given by,

$$\Delta\tau_{ij}^k = \begin{cases} \frac{1}{\lambda L_k + \gamma R_k} & \text{if } c_{ij} \in s^p \\ 0 & \text{otherwise} \end{cases}$$

V. EXPERIMENTAL SETUP AND FIELD TESTING

As discussed in Section III, We simulate the problem in a 2D environment, solve it using the the algorithm in Section IV, and then apply the solution to the real-life environment with drones. Figure 3 shows the farm for which we plan a coverage path, highlighting tall structures that the UAV path must avoid.



Fig. 3. Birds-eye image from Google Earth of the farm that the coverage path is designed for. The orange outer border is the boundary that the drones must stay within. The blue shapes surround tall obstacles on the farm that must be avoided. Blue circles surround tall trees. From left to right, the blue rectangles outline a house and greenhouse, respectively. The orange stars are the charging stations that the drones take off from and land on after searching the farm for deer.

A. Solving the EECPP problem in a 2D environment

We developed a 2D simulation environment that replicates key features of the agricultural landscape using VMAS [35]. The environment includes:

1) Waypoint Navigation: A grid of 36 waypoints is distributed across the simulated farm area, which is designed to guide drones for complete coverage. In our implementation, the points are 38 meters apart. The density of waypoints should be adjusted based on the fixed altitude of the drone during flight and the area of land that the drone is able to scan at any moment. We set up two types of problems: single-drone and dual-drone coverage path planning. While the farm is small enough to be surveilled by a single drone, we introduce the dual-drone problem as a proof of concept that could be applied to larger farms. For the single-drone problem, all waypoints are interconnected to achieve one complete graph as illustrated in Figure 1. For the dual-drone problem, the farm is divided vertically down the middle, so that one drone is responsible for all waypoints on the left, and the other is responsible for those on the right. After edge generation, we are left with two disjoint complete graphs.

2) Obstacle Avoidance: As illustrated in Figure 1, collision avoidance is implemented by removing edges that intersect with or are within a safety margin of any obstacle. This includes edges connected to invalid waypoints. This ensures that no coverage paths are generated that could cause a collision with obstacles included in the simulation.

3) Scaled Terrain Modeling: The simulation environment is designed to mirror real-world farm dimensions, with obstacles (such as a greenhouse, house, and trees) placed

according to the farms layout. As shown in Figure 4, the 2D top-down visualization renders these components to scale. Waypoints appear as green points, obstacles as red shaded shapes, and drones as purple circles.

In the dual-drone problem, each drone is assigned to separate waypoint regions to help prevent mid-mission collisions. However this does not guarantee that the UAVs' paths won't intersect, depending on where their charging stations are located; so, we also have UAVs fly at different altitudes [6] in the 3D implementation. The total distance traveled and total rotation are tracked in meters and degrees respectively so that the energy consumption of a path can be estimated using the energy cost estimation formula from Section IV.

The proposed system addresses three critical challenges in agricultural drone path planning: obstacle avoidance, multi-drone task partitioning, and energy efficiency. Solving the problem in a 2D environment serves as the foundation to solving it in a real-world deployment.

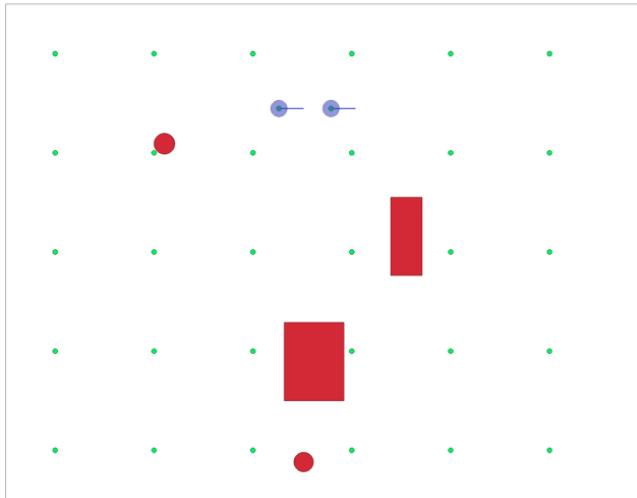


Fig. 4. 2D simulation of the UAV coverage path planning problem for the farm designed using VMAS [35]. The black outer border is the boundary that the drones must stay within. The green dots are the waypoints that the drone traverse. The red shapes are obstacles that the drones should avoid.

B. Applying 2D solution to real drones

To validate our 2D trajectory planning system in field deployments, the waypoints obtained from the simulation test bed are translated to georeferenced coordinates corresponding to the actual locations. The waypoints derived from the VMAS are mapped to contain the offsets and orientations, and scaled to fit the equivalent real-world coordinates that are uploaded into a QGround Control (QGC) [37] compatible file. As illustrated in Fig. 6, the UAVs are commanded to follow two distinct, closed-loop paths generated by the algorithm in Section IV. Also, the interface of Mavlink allows seamless automatic transfer of these converted waypoints to the UAV onboard computer, i.e., the Jetson Orion platform, thereby reducing the need for human intervention and easy mission updates.



Fig. 5. PX4 quadcopter that will be used in the implementation of the energy efficient coverage path planning algorithm for finding deer on the farm. It is equipped with a camera for autonomous landing, a camera for deer detection using a YOLO-based real-time computer vision module [36], an NVIDIA Jetson Orin Nano for onboard inference, and a speaker and lights for deterrence stimuli.

These field tests utilize UAVs of the PX4 6X flight controller, which have Real-Time Kinematic (RTK) GPS, Inertial Measurement Units (IMUs), and advanced collision detection sensors, combined to enable their great navigation under changing agricultural conditions. They have an onboard platform (Jetson Orion), flight control software, and Mavlink interface [38] to enable real-time mission deployment and adaptive operations, both of which accomplish precise waypoint navigation as well as responsiveness to changes in the environment.

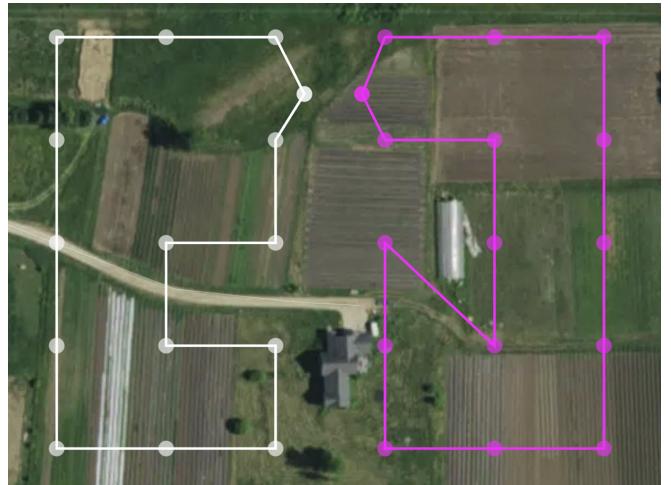


Fig. 6. Visualization of two closed-loop paths generated with ACO that are assigned to two UAVs. The white and pink paths are to be followed by their respective UAVs when searching for deer, ensuring complete coverage while avoiding obstacles.

C. Hyperparameter tuning

We used Optuna [39] for hyperparameter tuning of the ACO algorithms described in IV to find a combination of parameters that perform optimally in the dual-drone EECPP

problem. We configured it to maximize the sum of the heuristic functions of the tour of drone 1 and the tour of drone 2 over 40 trials. This is equivalent to minimizing the energy cost of the drones. Table I provides the parameters that were tuned, the ranges they were tuned for, and the final values that we used.

TABLE I
PARAMETER USED FOR THE ALGORITHM.

Parameter	Description	Range Tested	Finalized Values
n	Number of ants	[10,500]	414
$iter$	Number of epochs	[10,200]	138
ρ	Evaporation rate of pheromones	[0.001,1]	0.0134
α	Heuristic importance value	[0.1,5.0]	3.985
β	Pheromone importance value	[0.1,5.0]	2.3905
τ_{max}	Maximum pheromone value allowed on an edge	[2,10]	2.027
τ_{min}	Minimum pheromone value allowed on an edge	Constant	1
λ	Straight line energy cost (kJ/m)	Constant	0.1164
γ	Rotation energy cost (kJ/deg)	Constant	0.0173

The remaining parameters— τ_{min} , λ , and γ —were kept constant. We kept τ_{min} constant during the tuning process because what matters more than the actual values of τ_{min} and τ_{max} is the difference between them. This in combination with the computational resources that we saved during the training process, influenced our decision to leave τ_{min} constant and to tune τ_{max} . The values of λ and γ were set to the values from previous literature [28] that computed the energy costs of the maneuvers that we use in this paper for a Pixhawk drone: moving forward in a straight line and rotating (yaw). These values are only estimates of the energy consumption for the specific drones that we use. Calculating the energy consumption of maneuvering our specific drones would make the results of our application more accurate.

D. Baseline

We use a back-and-forth algorithm as the baseline for comparing against the ACO solutions. Our back-and-forth algorithm closely follows the algorithm used by Cao et al., [20]. We first draw a number of parallel “straight paths” through the waypoints, then repeatedly connect the two closest segments until a TSP tour is formed. Afterwards, we apply iterations of 2-opt until no further improvement is possible.

VI. RESULTS

Figures 7, 8, 9, and 10 show coverage paths for the farm in Figure 3. One waypoint is omitted in the top-left path due to proximity to a tall tree, represented by the circular landmark. In the dual-drone solutions (Figures 7 and 8), the farm is vertically split down the middle with each drone covering one side.

Table II summarizes the energy costs of the example MMAS (Figures 9 and 7) and back-and-forth (Figures 10 and 8) coverage paths for both single- and dual-drone scenarios. As explained in Section IV, the energy cost is given by $\lambda \cdot distance + \gamma \cdot rotation$, where the values of λ and γ

are provided in Table I. MMAS achieves a 12.8% energy reduction in the dual-drone case and a 4.9% reduction in the single-drone case when compared to the back-and-forth paths.

TABLE II
ENERGY COST COMPARISON OF MMAS AND BACK-AND-FORTH ALGORITHMS FOR UAV COVERAGE

Scenario	Method	Distance (m)	Rotation ($^{\circ}$)	Energy (kJ)
Dual-drone	MMAS	537.0 + 589.0	725.0 + 763.9	156.8
Dual-drone	Back-and-forth	616.6 + 713.6	740.7 + 700.5	179.8
Single-drone	MMAS	1179.2	1080.8	156.0
Single-drone	Back-and-forth	1246.9	1131.1	164.7

Figure 11 presents a comparative analysis of mean energy costs and performance metrics across the evaluated algorithms. The AS and MMAS algorithms were run for 30 trials, and the mean values of successful solutions—those in which all drones involved visit all valid waypoints and return to their charging stations—are included in the derivation of the mean values. For the dual-drone problem, 19 out of 30 trials of AS resulted in successful solutions, whereas 18 out of 30 trials of MMAS resulted in successful solutions. For the single-drone problem, 28 out of 30 trials of AS resulted in successful solutions, whereas 26 out of 30 trials of MMAS resulted in successful solutions. One trial is included for the back-and-forth algorithm as it inherently produces the same solution for a given problem space.

It is worth noting that we were able to get our version of the back-and-forth algorithm to work for the problem space by allowing slight deviation from the straight-line path between waypoints. This gives the back-and-forth algorithm an advantage over the ACO algorithms when comparing their results, as it can travel between waypoints that ACO does not have as options.

VII. DISCUSSION AND CONCLUSIONS

The results in Figure 11 show that for both the single- and dual-drone problems, AS and MMAS outperform the back-and-forth algorithm in terms of energy efficiency. Even when ACO-generated paths have more total rotation than the back-and-forth solutions (Turns chart in Figure 11), the ACO algorithms still generate more energy-efficient paths. This is possible because the energy consumption of a path is not determined solely by total rotation or distance, but a combination of the two. The degree to which distance and rotation effect energy consumption is dependent on the specific UAV being used. When comparing successful solutions of AS and MMAS, AS on average traveled slightly less distance and rotation, and thus had slightly better fuel efficiency.

A qualitative difference between the back-and-forth solution and the ACO solutions that we find particularly apparent in the single-drone coverage paths, as demonstrated in Figure 9, is that while the back-and-forth algorithm generally moves from one side of the farm to the next, the ACO algorithm adopts a more spiral flight pattern. The spiral flight pattern could be

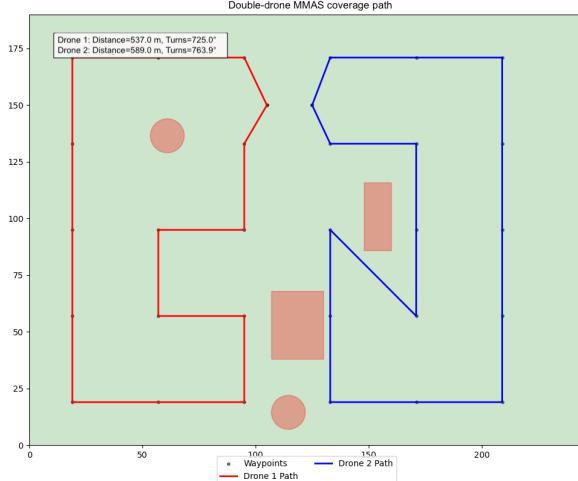


Fig. 7. Example paths of two drones covering the farm using the MMAS algorithm.



Fig. 9. Example path of one drone covering the farm using the MMAS algorithm.

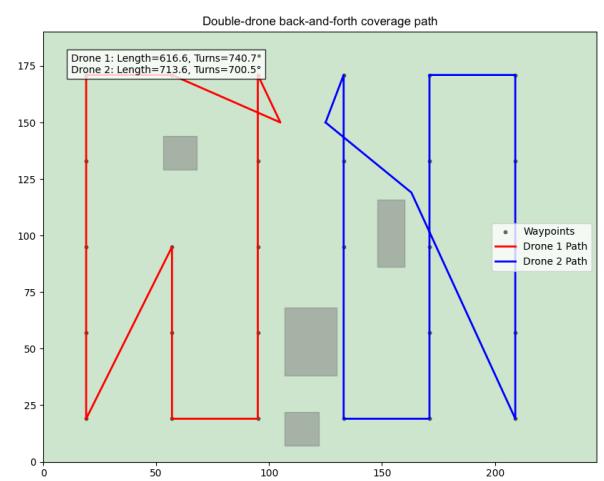


Fig. 8. Paths of two drones covering the farm using the back-and-forth algorithm.

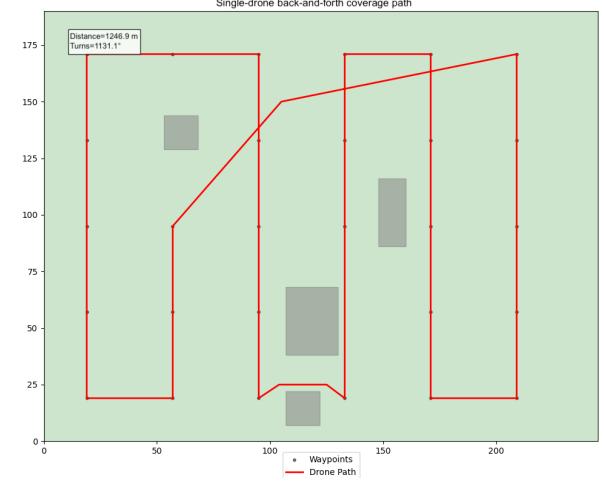


Fig. 10. Path of one drone covering the farm using the back-and-forth algorithm.

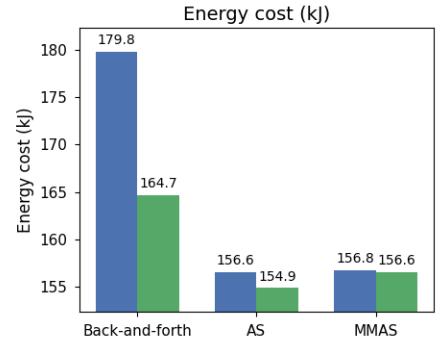
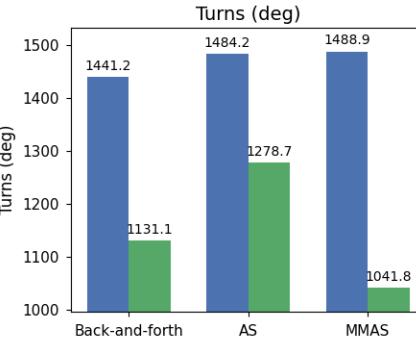
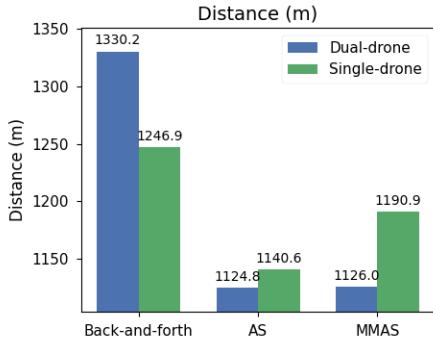


Fig. 11. Comparison of results of the back-and-forth, AS, and MMAS coverage planning algorithms for both the single-drone and double-drone problems. AS and MMAS are mean values of the valid solutions from 30 trials. For the dual-drone problem, 19 trials are included for AS and 18 out of 30 are included for MMAS. For the single-drone problem, 28 out of 30 trials are included for AS and 26 out of 30 trials are included for MMAS. One trial was conducted for back-and-forth since it always returns the same solution.

preferable as it may be more difficult for deer to predict and

avoid; however, experiments with deer need to be conducted

in order to confirm this intuition.

While AS and MMAS produced similar results when they were valid, MMAS converged on invalid solutions slightly more often. An invalid solution occurs when a path is unable to return to the charging station because there are no straight-line paths to waypoints that are untraversed. A reason why AS may be more likely to avoid merging on invalid solutions could be because AS explores more potential paths than MMAS does, since all ants deposit pheromones in AS, whereas only the best performing ant lays pheromones in MMAS. However, more trials need to be run on different problem spaces before we can say with certainty that AS performs better than MMAS in UAV EECPP problems.

In conclusion, in response to the negative impacts that wild deer have on farmers and their crops, we propose the use of UAVs to surveil farms as part of a deer deterrence system. We utilized ACO to generate an energy-efficient path planning algorithm that outperforms the commonly used back-and-forth algorithm in terms of energy efficiency so that the UAVs can cover a larger area. Our approach accounts for farms that are too large to be surveilled by a single UAV by partitioning the surveillance area and assigning them to different drones as necessary.

While our simulation results are promising, in-field performance and deer behavioral responses remain to be evaluated. More research is needed to determine whether deer are less able to predict the flight patterns generated by ACO and avoid detection while continuing to damage crops. Our current work focuses on deer detection, but future efforts could integrate a deterrence protocol for the UAVs to follow after detecting deer. Such a protocol could involve using multiple forms of deterrence stimuli, such as speakers and lights, and should consider how to address deer habituation.

In future work¹, we plan to incorporate our algorithm into a complete autonomous UAV system using the drone shown in Figure 5 and test its detection and deterrence capabilities on real deer. The drone is equipped with a speaker and lights, allowing us to experiment with different stimuli for deterrence. To improve the accuracy of our energy estimates, we will calculate the unit energy costs of traveling in a straight line and rotating based on our specific drone configuration and use these values in the ACO heuristic function. We also plan to compare the predicted energy cost of each path with the actual energy expenditure during flight. Additional improvements to the algorithm may include incorporating local search into our ACO implementation and allowing UAVs to deviate from the straight-line path between waypoints, as demonstrated by drone 2 in Figure 8. Finally, we will conduct a more thorough comparison between our algorithm and other state-of-the-art energy-efficient coverage path planning algorithms to better evaluate the effectiveness of our proposed system.

¹Up-to-date information about our project and implementation is available at <https://ebasatemesgen.github.io/FarmGuard/>

ACKNOWLEDGMENT

We thank Chris and Susan James, owners of Fresh Earth Farms for working closely with us on this project and allowing us to base our research on their farm. We also thank Greta Brown for assisting with the 2D environment and Risako Owan for providing valuable feedback on presentation style.

REFERENCES

- [1] W. M. Sarmento, "Drones outperform dogs for hazing bears: a comparison of carnivore aversive conditioning tools," *Frontiers in conservation science*, vol. 5, 2025, publisher: Frontiers Media S.A. [Online]. Available: <https://www.frontiersin.org/journals/conversation-science/articles/10.3389/fcosc.2024.1478450/full>
- [2] F. Schiano, D. Natter, D. Zambrano, and D. Floreano, "Autonomous detection and deterrence of pigeons on buildings by drones," *IEEE Access*, vol. 10, pp. 1745–1755, 2022.
- [3] Z. Wang and K. C. Wong, "Autonomous pest bird deterring for agricultural crops using teams of unmanned aerial vehicles," in *2019 12th Asian Control Conference (ASCC)*, 2019-06-09, pp. 108–113, journal Abbreviation: 2019 12th Asian Control Conference (ASCC).
- [4] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.
- [5] T. Kennedy, "Minnesota hunters take aim at program allowing farmers to shoot large numbers of deer," *TCA News Service*, 2024.
- [6] Z. Wang and K. Wong, "Autonomous bird deterrent system for vineyards using multiple bio-inspired unmanned aerial vehicle," in *Proc. 10th Int. Micro Air Vehicle Conf. Competition (IMAV)*, vol. 2018, 2018, pp. 256–281.
- [7] W. M. Sarmento, "Bear deterrence with scare devices, a non-lethal tool in the use-of-force continuum," *The Journal of wildlife management*, vol. 88, no. 3, p. n/a, 2024, place: Bethesda Publisher: Blackwell Publishing Ltd.
- [8] T. W. Logan, A. Ashton-Butt, and A. I. Ward, "Improving daytime detection of deer for surveillance and management," *European Journal of Wildlife Research*, vol. 65, no. 6, p. 83, 2019-10-14. [Online]. Available: <https://doi.org/10.1007/s10344-019-1318-y>
- [9] M. A. C. Rodriguez, "Deer surveillance system in public parks using deep learning," Master's thesis, National College of Ireland, 2022.
- [10] A. Sudholz, "Machine learning for the automated detection of deer in drone and camera trap imagery," mastersthesis, Queensland University of Technology, 2021. [Online]. Available: <https://eprints.qut.edu.au/212981/>
- [11] M. Israel, "A UAV-based roe deer fawn detection system," *International archives of the photogrammetry, remote sensing and spatial information sciences.*, vol. XXXVIII-1/C22, pp. 51–55, 2012, publisher: Copernicus Publications.
- [12] K. Bezas, G. Tsoumanis, and K. Oikonomou, "A coverage path planning algorithm for self-organizing drone swarms," in *2021 International Balkan Conference on Communications and Networking (BalkanCom)*, 2021-09-20, pp. 122–126, journal Abbreviation: 2021 International Balkan Conference on Communications and Networking (BalkanCom).
- [13] E. V. Vazquez-Carmona, J. I. Vasquez-Gomez, J. C. Herrera-Lozada, and M. Antonio-Cruz, "Coverage path planning for spraying drones," *Computers & Industrial Engineering*, vol. 168, p. 108125, 2022-06-01. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0360835222001954>
- [14] X. Mu, W. Gao, X. Li, and G. Li, "Coverage path planning for UAV based on improved back-and-forth mode," *IEEE Access*, vol. 11, pp. 114 840–114 854, 2023. [Online]. Available: <https://ieeexplore.ieee.org/document/10287347>
- [15] R. I. Mukhamiediev, K. Yakunin, M. Aubakirov, I. Assanov, Y. Kuchin, A. Symagulov, V. Levashenko, E. Zaitseva, D. Sokolov, and Y. Amirgaliyev, "Coverage path planning optimization of heterogeneous UAVs group for precision agriculture," *IEEE Access*, vol. 11, pp. 5789–5803, 2023.
- [16] M. Torres, D. A. Pelta, J. L. Verdegay, and J. C. Torres, "Coverage path planning with unmanned aerial vehicles for 3d terrain reconstruction," *Expert Systems with Applications*, vol. 55, pp. 441–451, 2016-08-15. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417416300306>

- [17] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," in *Field and Service Robotics*, A. Zelinsky, Ed. Springer London, 1998, pp. 203–209.
- [18] H. Choset, E. Acar, A. A. Rizzi, and J. Luntz, "Exact cellular decompositions in terms of critical points of morse functions," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 3, 2000-04-24, pp. 2270–2277 vol.3, journal Abbreviation: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065).
- [19] Y. Li, H. Chen, M. Joo Er, and X. Wang, "Coverage path planning for UAVs based on enhanced exact cellular decomposition method," *Special Issue on Development of Autonomous Unmanned Aerial Vehicles*, vol. 21, no. 5, pp. 876–885, 2011-08-01. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957415810001893>
- [20] Y. Cao, X. Cheng, and J. Mu, "Concentrated coverage path planning algorithm of UAV formation for aerial photography," *IEEE Sensors Journal*, vol. 22, no. 11, pp. 11 098–11 111, 2022-06-01.
- [21] E. L. Lawler, "The traveling salesman problem: a guided tour of combinatorial optimization," *Wiley-Interscience Series in Discrete Mathematics*, 1985.
- [22] G. R. Gallagher, J. L. Peacock, E. P. Garner, and R. H. Prince, "Conditioning and habituation of white-tailed deer to two common deterrents," in *Wildlife Damage Management Conference Proceedings*, 2000.
- [23] Y. Espmark and R. Langvatn, "Development and habituation of cardiac and behavioral responses in young red deer calves (*cervus elaphus*) exposed to alarm stimuli," *Journal of Mammalogy*, vol. 66, no. 4, pp. 702–711, 1985-11-29. [Online]. Available: <https://doi.org/10.2307/1380796>
- [24] C. Di Franco and G. Buttazzo, "Coverage path planning for uavs photogrammetry with energy and resolution constraints," *Journal of Intelligent & Robotic Systems*, vol. 83, pp. 445–462, 2016.
- [25] C. Zhao, Y. Liu, and J. Zhao, "Path planning method of uav area coverage searching based on pega," *Science & Technology Review*, vol. 32, no. 28/29, pp. 85–90, 2014.
- [26] Z. Chibin, W. Xingsong, and D. Yong, "Complete coverage path planning based on ant colony algorithm," in *2008 15th International Conference on Mechatronics and Machine Vision in Practice*. IEEE, 2008, pp. 357–361.
- [27] J. Chen, F. Ling, Y. Zhang, T. You, Y. Liu, and X. Du, "Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system," *Swarm and Evolutionary Computation*, vol. 69, p. 101005, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S221065022100167X>
- [28] J. Modares, F. Ghanei, N. Mastronarde, and K. Dantu, "UB-ANC planner: Energy efficient coverage path planning with multiple drones," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017-06-29, pp. 6182–6189.
- [29] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proceedings of the first European conference on artificial life*, vol. 142. Paris, France, 1991, pp. 134–142.
- [30] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation, Politecnico di Milano, 1992.
- [31] V. Murugananthan, M. Y. E. S. Rehan, R. Srinivasan, M. Kavitha, and R. Kavitha, "Traveling salesman problem with ant colony optimization," in *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*, 2023-07-19, pp. 481–485, journal Abbreviation: 2023 2nd International Conference on Edge Computing and Applications (ICECAA).
- [32] M. Dorigo and L. M. Gambardella, "Ant colonies for the traveling salesman problem," *Biosystems*, vol. 43, no. 2, p. 73–81, 1997.
- [33] T. Stutzle and H. Hoos, "MAX-MIN ant system and local search for the traveling salesman problem," in *ICEC*. IEEE, 1997, pp. 309–314, journal Abbreviation: ICEC.
- [34] T. Stützle and H. H. Hoos, "MAX-MIN ant system," *Future generation computer systems*, vol. 16, no. 8, pp. 889–914, 2000.
- [35] M. Bettini, R. Kortvelesy, J. Blumenkamp, and A. Prorok, "VMAS: A vectorized multi-agent simulator for collective robot learning," 2022. [Online]. Available: <https://arxiv.org/abs/2207.03530>
- [36] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016. [Online]. Available: <https://arxiv.org/abs/1506.02640>
- [37] C. Ramirez-Atencia and D. Camacho, "Extending qgroundcontrol for automated mission planning of uavs," *Sensors*, vol. 18, no. 7, p. 2339, 2018.
- [38] A. Koubaâa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui, "Micro air vehicle link (mavlink) in a nutshell: A survey," *IEEE Access*, vol. 7, pp. 87 658–87 680, 2019.
- [39] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 2623–2631. [Online]. Available: <https://doi.org/10.1145/3292500.3330701>