

Article

An Algorithm for Planning Coverage of an Area with Obstacles with a Heterogeneous Group of Drones Using a Genetic Algorithm and Parameterized Polygon Decomposition

Kirill Yakunin ^{1,2} , Yan Kuchin ^{1,3} , Elena Muhamedijeva ^{1,3}, Adilkhan Symagulov ^{1,3,*} and Ravil I. Mukhamediev ^{1,3} 

¹ Institute of Automation and Information Technologies, Satbayev University (KazNRTU), Almaty 050013, Kazakhstan; yakunin.k@mail.ru (K.Y.); ykuchin80@gmail.com (Y.K.); muhamedijeva@gmail.com (E.M.); ravil.muhamedyev@gmail.com (R.I.M.)

² School of Digital Technologies, Almaty Management University, Rozybakiev Street 227, Almaty 050060, Kazakhstan

³ Institute of Information and Computational Technologies, Pushkin Str., 125, Almaty 050013, Kazakhstan

* Correspondence: asmogulove00@gmail.com

Abstract

The paper presents an algorithm for planning agricultural field surveying routes in the presence of obstacles, designed to address precision agriculture tasks. Unlike classical methods, which are typically limited to straightforward zigzag (Zamboni) traversal and basic perimeter-based obstacle avoidance, the proposed algorithm accounts for heterogeneous unmanned aerial vehicles (UAVs) of varying types, ranges, costs, and speeds, along with a mobile ground platform that enables drone takeoff and landing at multiple points along the road. The key innovation lies in a two-stage optimization procedure: initially, a random set of field partitions into multiple sub-polygons with predefined area proportions (considering internal obstacles) is generated. Subsequently, the optimal partitioning is selected, and based on this, a genetic algorithm is applied to optimize flight parameters, including flight angle, entry points, composition, and sequence of drone launches, and the ground platform route. This approach achieves more localized coverage of individual field segments, with each segment serviced by an appropriate drone type, while also enabling flexible movement of the ground platform, thereby reducing unnecessary flights. This brings down the price of the coverage by 10–30% in some cases. The concluding section discusses future directions, including the incorporation of three-dimensional terrain considerations, dynamic factors (such as changing weather conditions and drone stoppages due to technical issues), and automated collision avoidance in intersecting route segments.

Keywords: coverage algorithm; unmanned aerial vehicles; genetic algorithm; flight planning; artificial intelligence



Academic Editors: Haokui Zhang, Liang Wang, Jingtao Sun and Xizhe Xue

Received: 11 July 2025

Revised: 4 September 2025

Accepted: 12 September 2025

Published: 18 September 2025

Citation: Yakunin, K.; Kuchin, Y.; Muhamedijeva, E.; Symagulov, A.; Mukhamediev, R.I. An Algorithm for Planning Coverage of an Area with Obstacles with a Heterogeneous Group of Drones Using a Genetic Algorithm and Parameterized Polygon Decomposition. *Drones* **2025**, *9*, 658. <https://doi.org/10.3390/drones9090658>

Correction Statement: This article has been republished with a minor change. The change does not affect the scientific content of the article and further details are available within the backmatter of the website version of this article.

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Modern precision agriculture technologies are based on a differentiated approach to crop cultivation and management. Unlike traditional methods, where fertilizers, herbicides, or other agricultural inputs are uniformly applied across an entire field, precision agriculture involves collecting detailed data on the condition of each specific area (for example, using optical, thermal, or multispectral sensors mounted on unmanned aerial vehicles—UAVs) and making localized decisions for resource application. This enhances

the efficiency of fertilizer and chemical usage, reduces adverse impacts on the soil, and optimizes energy consumption.

However, to ensure maximally effective data collection and targeted substance application, UAV flight paths must be planned in such a way as to comprehensively “cover” (i.e., fly over) all relevant areas of the field with the required overlap. Coverage path planning (CPP) in precision agriculture contexts presents several challenges:

1. Complex field geometries. Agricultural fields are often non-rectangular and may exhibit non-convex shapes (such as concave indentations) or other geometric peculiarities.
2. Presence of obstacles within the field that are either inaccessible for flight or require no treatment (lakes, rocky areas, tall tree belts, farmyards). Additionally, obstacle areas may represent regions to be excluded from coverage (for instance, areas that do not require herbicide spraying). In this work, “obstacles” refer to zones within field boundaries that must be entirely excluded from flight paths—either because flying there is impossible or impractical.
3. Heterogeneity of the UAV fleet. Different UAV models (multicopters, helicopter-type drones, fixed-wing aircraft) may exist within a single agricultural operation, each with distinct operational ranges, speeds, operational costs, payload capacities, and other characteristics.
4. Use of a mobile ground platform (tractor, truck, pickup) that moves along the road at the edge of the field or traverses it via feasible paths. Such a platform enables more efficient logistics: drones can take off and land at various locations along the road instead of being limited to a single point, reducing total drone mileage and turnaround times for battery replacement or refueling.

To integrate all these aspects, a comprehensive solution is required, enabling the system to do the following:

- Account for geometry: complex field shapes and obstacles.
- Optimally allocate involved UAVs (considering their differing specifications).
- Plan an optimal route (or set of stops) for the ground platform to minimize overall mission duration.
- Reduce costs and time (two key metrics): where cost encompasses the total expenditure related to UAV wear and tear, depreciation, and personnel wages; and time refers to the duration of the entire operation, typically constrained by the working day or favorable weather window.

In this paper, we propose a method based on a genetic algorithm (GA) combined with an additional “partitioning” of the field (excluding obstacles) into multiple segments, each constituting a defined portion of the total area. This approach is potentially more flexible and cost-effective compared to the basic approach, in which the entire field (without explicit partitioning) is traversed using a standard Zamboni route (zigzag or back-and-forth pattern), with obstacles circumvented in either a clockwise or counterclockwise direction. The basic approach and its inherent limitations will be examined in detail, followed by the presentation of our improved two-stage methodology.

The paper is structured as follows:

Section 2 provides a literature review of the current state of UAV path planning problems.

Section 3 describes the method proposed for solving the coverage problem.

Section 4 presents and briefly discusses the obtained results.

In the conclusion, we summarize findings, outline the limitations of the current research stage, and propose future research directions.

2. Related Work

With the development and increased intelligence of UAV technologies, the number of their applications across various domains is steadily growing. UAV technologies are utilized to enhance efficiency in agriculture [1], mining operations and mineral exploration [2], magnetic surveying of volcanoes [3], establishment of emergency communication networks [4], monitoring urban agglomerations [5], forest fires [6], wildlife [7], marine ecosystems [8], water bodies [9], engineering structures [10–12], road traffic [13], and environmental pollution [5,14], among numerous other fields. A generalized scheme of UAV applications derived from the analysis of relevant publications [15–18] is presented in Figure 1.

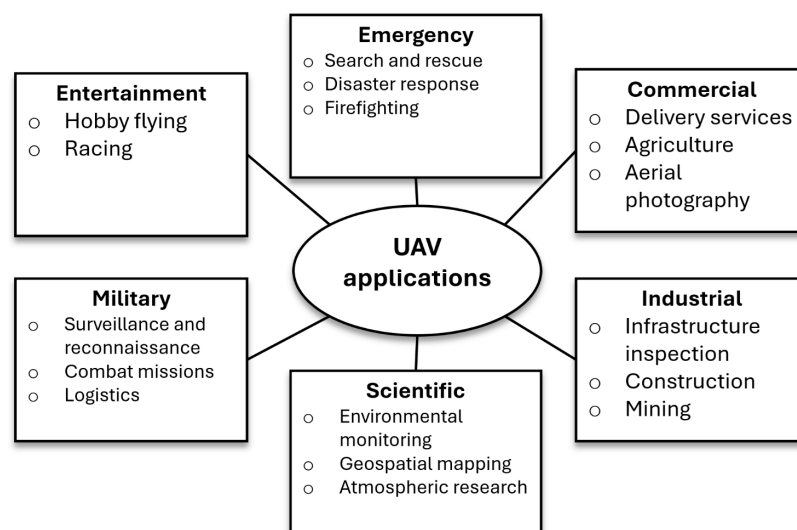


Figure 1. Applications of UAV technology.

Improving the degree of automation in the aforementioned UAV application scenarios necessitates advancements in control methods and data processing, considering the inherent advantages and limitations of UAVs. As technical devices, UAVs have several significant limitations in practical deployment. These include technical constraints (limited flight duration, dependence on weather conditions and satellite navigation systems, etc.) [19], legal constraints (prohibitions or restrictions related to UAV flights based on their mass) [20], and software–algorithmic challenges (control under complex conditions and processing of large volumes of high-resolution imagery) [21,22]. Overcoming these limitations in practical applications requires thorough mission planning, not only for individual drones but also for UAV groups solving a shared task with the support of ground assets. Consequently, the UAV flight planning problem has attracted considerable research attention in various implementations, including cargo transportation [23], control under external disturbances [24], UAV transfer between moving maritime vessels [25], swarm coordination [26,27], collaborative missions [28], search operations [29], surveillance tasks [30], and chaotic movement in hostile environments [31], among others.

One variant of flight planning is coverage path planning (CPP), commonly applied in various monitoring tasks [32]. For single-UAV control, this task has largely been resolved and incorporated into several flight-controller management software solutions (QGround Control [33], Dronekit [34], ArduPilot [35], Universal Ground Control Software [36], DJI Mavic 3 Enterprise, DJI Mavic 3 Multispectral). However, the problem of controlling multiple UAVs under heterogeneous constraints, in cooperation with flight-support systems (charging stations, vehicles, personnel), continues to be an active research focus. Several studies are dedicated to this topic. For instance, the shortest-path planning problem within

vineyard hedgerow systems in Spain is investigated in [37]. The task of 3D trajectory planning for multiple fixed-wing UAVs is discussed in [38]. Reference [39] considers coverage path planning with energy constraints and obstacles for oil and gas exploration purposes. The coverage path planning problem involving robots navigating pedestrian scenarios is analyzed in [40], with emphasis on obstacle avoidance within zones of interest. Multi-region coverage path planning for agricultural machinery is examined in [41]. The coverage algorithm for pesticide spraying using agricultural UAVs is described in [42], targeting flight range and coverage ratio as primary optimization objectives. Special path planning methods were used to handle concave polygonal regions. Reference [43] proposes a solution for spot spraying weed patches obstructed by trees and water bodies. This method integrates a heuristic solution to the Traveling Salesman Problem (TSP) with optimized coverage path planning, with the coverage path being the largest contributor to overall path length.

In real-world scenarios, the path planning problem is inherently multi-objective, resulting in complex objective functions that make classical linear or nonlinear optimization methods challenging. Therefore, various heuristic optimization algorithms have been employed for coverage path planning tasks, including ant colony optimization [44], the Green Anaconda algorithm [45], machine learning algorithms, fuzzy logic, swarm optimization [46], evolutionary programming [47], and notably, various modifications of genetic algorithms [48,49]. One critical limitation of existing algorithms and methods is the lack of consideration given to UAV fleet heterogeneity. This limitation is addressed in [50], where the authors propose a solution to the problem of multi-heterogeneous UAV coverage path planning with a moving ground platform (mhCPPmp) for large agricultural fields. Moreover, the optimization objective in this referenced work is minimizing the total mission cost rather than merely reducing either path length or energy consumption.

Recent work also adopts hierarchical/two-stage schemes for multi-UAV coordination in logistics settings. Wen et al. propose a bi-layer collaborative framework that couples structured 3D environment modeling with multi-depot task assignment for urban delivery, showing that separating strategic resource decisions from tactical routing improves tractability under depot and resource constraints [51]. Complementarily, Zhang et al. present a two-stage optimization in which a metaheuristic first determines fleet size and a decentralized CBBA then handles task allocation under a comprehensive cost objective [52]. Methodologically, both support our design choice to decouple partition selection from route optimization; unlike our work, they target delivery rather than area coverage and do not model a moving ground platform or polygon-decomposition-based coverage with internal obstacles.

The present work, in a sense continuing this line of research, addresses the problem of coverage path planning using multiple heterogeneous UAVs in the presence of obstacles and a mobile ground-based flight-support platform.

3. Method

3.1. Baseline Approach to Coverage: Zamboni with Obstacle Avoidance

To understand the contribution of the proposed method, we first consider the classical and straightforward solution, referred to here as the baseline approach. Its working principle is described below:

- (1) Zamboni route (zigzag traversal).

The field is covered by parallel passes: the drone flies along a straight path, then makes a turn and moves back slightly offset from the previous stripe. This resembles the ice resurfacing pattern on hockey rinks or the operation of harvesting combines. Turns are usually performed at the field edges.

- (2) Obstacle avoidance (clockwise or counterclockwise).
If the field contains a substantial obstacle area (e.g., a pond or tree belt), the classical method formally “cuts out” the obstacle contour from its route, circumnavigating it either clockwise or counterclockwise before resuming straight-line movement. This can introduce lengthy maneuvers around obstacles, creating additional turns and unnecessary segments. Practically, each obstacle encounter can split a single straight stripe into three or four separate segments.
- (3) Single drone or multiple identical drones.
Typically, CPP tasks assume either a single drone or multiple identical drones (with uniform characteristics), all taking off from a single location. If refueling (or recharging) is required, the drone returns to the same point, then resumes the route from the point of interruption.
- (4) No ground-platform route optimization.
Existing studies do not usually account for the potential use of a mobile platform. It is assumed to be stationary (or absent entirely), meaning there is no mechanism to shorten the drone’s path by moving the platform to points where the drone needs to land.

The primary limitation of this approach is its lack of flexibility in obstacle traversal. When building a Zamboni route, the angle at which the drone enters the area must be chosen and fixed. Depending on this chosen angle, all obstacles—regardless of their position or shape—are circumnavigated uniformly. Later in the article, scenarios will be presented demonstrating that such an approach does not yield optimal flight plans. Consequently, overall time and costs may be suboptimal, primarily due to redundant flights over the same territory during multiple takeoff–landing cycles and inefficient obstacle avoidance. Nonetheless, this method is straightforward to implement—many commercial drone route-planning software solutions operate exactly in this manner, with minimal or no obstacle specification.

This study proceeds from the hypothesis that further subdividing the field into sub-polygons (where each polygon can be efficiently covered by a specific drone type or at a particular angle), combined with advanced platform route optimization, potentially improves economic performance—this is particularly important for large, complex fields.

3.2. Concept of Coverage Algorithm with Obstacle Avoidance

The proposed flight planning algorithm is based on the following ideas:

- (1) Field division (with obstacles) into fractional zones:
A field containing obstacles can be represented as an original polygon P minus the internal restricted areas (obstacles).
Instead of covering the entire polygon with one continuous Zamboni route, we subdivide the field into subsets $P_1, P_2, \dots, P_m$ so that they do not overlap and collectively cover the entire available area. Each P_i is assigned a fraction of the total area (for example, 30%, 20%, 50%, etc.).
The rationale behind this subdivision is that certain areas (sub-polygons) might be more conveniently covered by tiltrotors (efficient for long, straight lines), while other areas (winding or fragmented) are better suited for multicopters. Additionally, subdividing the field into multiple blocks can help minimize the overall number of obstacle fly-arounds. This subdivision is performed using the *pode* library [53], implementing the algorithm from [54].
- (2) Two-stage optimization approach:
Simultaneous optimization of polygon subdivision and flight parameters is infeasible, as changing the polygon subdivision (thus altering the overall zigzag shape) can

render previously optimal flight parameters suboptimal. Thus, these two components must be optimized separately. The space of possible polygon subdivisions is continuous, with numerous possible variations; therefore, a discrete approach involving the separate optimization of flight parameters for every potential subdivision would be computationally expensive.

Consequently, we propose a two-stage optimization approach:

- In the first stage, multiple subdivision configurations are generated (each configuration represented by an area fraction vector summing to 1). For each subdivision, preliminary economic efficiency is evaluated through a set of random flight plans. The subdivision yielding minimal flight cost is selected as potentially optimal.
- In the second (main) stage, a genetic algorithm is employed: candidate solutions encode flight routes (including flight angle, entry point, drone assignment and sequence, mobile platform trajectory) based on the optimal polygon subdivision identified in the first stage. Iteratively selecting and combining candidates, the algorithm searches for an optimal route for the given polygon configuration excluding obstacles.

Figure 2 shows the visual representation of the two-stage approach described above.

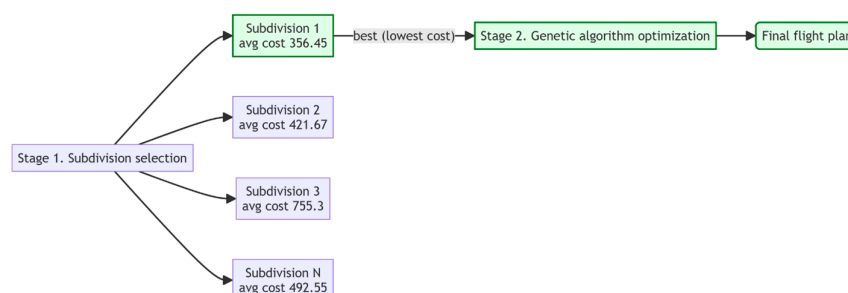


Figure 2. Visual representation of the two-stage approach. Green blocks represent subdivision with the lowest cost used in the Stage 2 of the approach. Blue blocks represent other subdivisions that were discarded due to higher average cost.

Genetic algorithms are metaheuristics based on stochastic selection methods that do not guarantee finding a global optimum but allow a solution close to the optimal one to be obtained. In this context, “optimal route” or “optimal decision” does not mean a global optimal solution in the strict sense of the term, but only a solution close to the optimal one.

3.3. Detailed Algorithm Description

Below is a comprehensive description of the entire process. Input data includes the following:

- Field polygon P ,
- Set of obstacles $\{H_1, H_2, \dots, H_s\}$, each $H_i \subset P$,
- Road (or roads) D , along which the ground platform moves,
- Set of drones $L = \{L_1, \dots, L_l\}$ with defined characteristics,
- Economic parameters: pilot hourly rate, takeoff fee, drone depreciation costs, etc.

Algorithm:

Compute the available field area considering obstacles:

$$A_{available} = Area(P) - \sum_{i=1}^s Area(H_i) \quad (1)$$

Assume obstacles do not intersect.

Stage 1 (Preliminary optimal subdivision search):

- (1) Select parameter N —number of partition vectors to test. Each subdivision consists of random fractions summing to 1.
- (2) For n from 1 to N :
 - Generate random vector $r = (r_1, \dots, r_m)$, with $\sum r_i = 1$ and random dimensionality m .
 - Partition field into subsets $\{P_1, \dots, P_m\}$, each with area $P_i = r_i \times A_{\text{available}}$ using poode library.
 - Generate M random flight configurations—flight angle, start point, drones, and landing points.
 - Estimate total cost for each configuration (without full GA execution).
 - Compute cost metrics (average + minimum).
- (3) Select subdivision r^* yielding best cost metrics.

Stage 2 (Genetic optimization based on best subdivision):

- (1) Input: optimal subdivision r^* .
- (2) Initialize GA population: each candidate encodes the following:
 - θ —flight angle ($0 \dots 360^\circ$);
 - start_corner—one of («NW», «NE», «SW», «SE»);
 - dronesList—ordered drone indices;
 - carPoints—landing points along road (0 to 1 proportional positions);
 - subPolygonsTraversalOrder—traversal order of sub-polygons r^* .
- (3) Each GA generation includes the following:
 - Decoding: We convert each individual into a real-world route. Considering the subPolygonsTraversalOrder and angle θ , zigzag flight paths are constructed. Drone missions are simulated using the specified dronesList, takeoff and landing points from carPoints, and the initial start_corner position.
 - Cost evaluation: We calculate operational expenses (flight costs, takeoff charges, and operator salaries) and apply a penalty P_{penalty} if the total mission time T_{total} exceeds a predefined threshold.
 - Selection: Individuals are sorted based on their objective value (total cost). The top-ranking individuals participate in crossover.
 - Crossover: We randomly select pairs of individuals to exchange portions of their genomes (for instance, flight angle, part of drone order list, etc.).
 - Mutation: With a certain probability, we adjust the flight angle, reorder drones, add or remove points in carPoints, etc.
- (4) Upon reaching the stopping criterion (either a sufficient number of generations or convergence of results), the individual with the best objective value is selected. This route is then used as the final coverage plan.

3.4. Formal Definition of the Objective Function

Route optimization using a genetic algorithm is performed based on an objective function incorporating multiple interrelated economic and time criteria. Formally, the objective function F_{total} for an individual x is defined as follows:

$$F_{\text{total}} = C_{\text{drone}}(x) + C_{\text{salary}}(x) + P(x) \quad (2)$$

where

- $C_{\text{drone}}(x)$ —total operational costs of the UAV fleet (USD);
- $C_{\text{salary}}(x)$ —total wages for drone operators (pilots) (USD);

- $P(x)$ —penalty function for violations of time or quality constraints (USD).

Calculation of UAV operational expenses:

$$C_{drone}(x) = \sum_{i=1}^n \left(p_i^{cycle} + p_i^{km} * d_i + p_i^{hour} * t_i \right) \quad (3)$$

where

- p_i^{cycle} —fixed cost per takeoff–landing cycle for drone i (USD);
- p_i^{km} —cost per kilometer of flight for drone i (USD/km);
- p_i^{hour} —hourly operational cost for drone i (USD/h);
- d_i —total distance traveled by drone i (km);
- t_i —total flight duration of drone i (hours).

The fixed cost per drone takeoff is justified not only by labor but also by increased risks associated with equipment wear, battery stress, and mechanical component strain during takeoff and landing, causing quicker depreciation compared to steady flight conditions.

The division of operational expenses into hourly and per-kilometer components reflects different depreciation factors and maintenance costs. Hourly costs account for time-dependent expenses (e.g., energy consumption, battery use, motor resources), while per-kilometer costs reflect component wear that is directly proportional to distance traveled (e.g., propeller lifespan, structural elements, mechanical stress).

Calculation of operator salaries:

$$C_{salary}(x) = (T(x) * h_{price}) + (n_{starts} * s_{price}) \quad (4)$$

where

- $T(x)$ —maximum flight duration among all drones (hours);
- h_{price} —operator's hourly wage rate (USD);
- n_{starts} —total number of drone takeoffs;
- s_{price} —fixed operator payment per takeoff (USD).

Using two cost components for the operator—hourly rate h_{price} and takeoff payment s_{price} —reflects different levels of personnel workload. The hourly rate covers time spent operating and monitoring missions, while the fixed takeoff payment acknowledges the increased workload and additional personnel (e.g., assistants or loaders), as each takeoff and landing requires heightened attention, increases risks, and accelerates drone wear.

Penalty function

Penalty function $P(x)$ ensures compliance with time and coverage constraints:

$$P(x) = \begin{cases} \left(C_{salary}(x) + C_{drone}(x) \right) * 10^{\min(T(x)-T_{max}, 10)}, & \text{если } T(x) > T_{max} \\ \left(C_{salary}(x) + C_{drone}(x) \right) * \frac{T(x)-T_{border}}{T_{max}-T_{border}}, & \text{if } T_{border} < T(x) \leq T_{max} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Additionally, if field coverage is insufficient (more than three waypoints missed), an extra penalty of 1,000,000 USD is imposed.

Parameters T_{max} and T_{border} are set to ensure timely task completion and manage risks related to exceeding the allocated mission time window. Intuitively, T_{border} represents the time threshold after which penalties begin to accumulate, signaling the mission is approaching critical deadlines but is still permissible. If the actual mission time exceeds T_{max} , it indicates serious planning issues, triggering exponentially growing penalties. Thus, these parameters incentivize the algorithm to select solutions guaranteeing stable execution within a comfortable time interval.

The additional penalty for missing waypoints ensures high-quality coverage of the required area. Missing waypoints can negatively impact precision agricultural activities, such as uneven fertilizer distribution or incomplete data collection, adversely affecting results. The high penalty enforces strict adherence to comprehensive area coverage.

Consequently, the proposed objective function provides a comprehensive approach to economic and operational mission planning, accounting for real-world operational conditions, risks, and drone fleet and personnel costs. It facilitates cost minimization while enhancing the reliability and practical feasibility of the obtained solutions.

3.5. Pseudocode of the Generalized Algorithm

Below is the pseudocode of the Generalized Algorithm—Algorithm 1 (language-agnostic, based on ISO/IEC 5807 standard):

Algorithm 1: TWO_STAGE_GA_FOR_HETEROGENEOUS_UAV_COVERAGE

INPUT:

```
P                // field polygon
OBSTACLES = {H1...Hs}
ROAD            // drivable path(s) for ground platform
L              // set of drones with parameters
N, M, MAXGEN    // search counts and GA generations
T_MAX, T_BORDER // time thresholds for penalties
COST_PARAMS     // per-cycle, per-km, per-hour, salary, per-takeoff,
etc.
```

OUTPUT:

```
BEST_PLAN        // final coverage plan
```

```
PROCEDURE TWO_STAGE_GA(P, OBSTACLES, ROAD, L, N, M, MAXGEN, T_MAX,
T_BORDER, COST_PARAMS) RETURNS BEST_PLAN
```

```
  A_AVAIL <- AREA(P) - SUM_OVER(i=1...s){ AREA(Hi) }
```

```
  BEST_SCORE <- +INFINITY
```

```
  BEST_PART <- NULL
```

```
  // ----- STAGE I: PARTITION SEARCH -----
```

```
  FOR n FROM 1 TO N DO
```

```
    R <- RANDOM_PARTITION_VECTOR()           // dimension m chosen at
random, SUM R[i] = 1
```

```
    PART <- DIVIDE_WITH_OBSTACLES(P, OBSTACLES, R) // e.g., via PODE;
produces {P1...Pm}
```

```
    SCORES <- EMPTY_LIST()
```

```
    FOR t FROM 1 TO M DO
```

```
      PLAN <- RANDOM_PLAN() // THETA, START_CORNER, DRONES_LIST,
CAR_STOPS, SUBPOLY_ORDER
```

```
      COST_EST <- QUICK_EVALUATE(PLAN, PART, ROAD, L, COST_PARAMS)
```

```
      APPEND(SCORES, COST_EST)
```

```
    END FOR
```

Algorithm 1. *Cont.*

```

    METRIC <- AGGREGATE(SCORES) // e.g., MIN(SCORES) + ALPHA *
    MEAN(SCORES)
    IF METRIC < BEST_SCORE THEN
        BEST_SCORE <- METRIC
    BEST_PART <- PART
    END IF
    END FOR

    // ----- STAGE II: GENETIC OPTIMIZATION -----
    POP <- INITIALIZE_POPULATION()
    FOR g FROM 1 TO MAXGEN DO
        FOR EACH IND IN POP DO
            ROUTE <- DECODE(IND, BEST_PART, ROAD, L) // build sweeps by THETA
            and SUBPOLY_ORDER; assign launches/landings
            IND.FITNESS <- EVALUATE_COST(ROUTE, T_MAX, T_BORDER, COST_PARAMS)
        END FOR

        POP <- SELECT(POP) // e.g., tournament or rank selection
        POP <- CROSSOVER(POP) // exchange substrings: angle, drone order,
        stop subsets, etc.
        POP <- MUTATE(POP) // perturb angle, reorder drones, add/remove car
        stops, etc.
    END FOR

    BEST_PLAN <- ARGMIN_FITNESS(POP)
    RETURN BEST_PLAN
END PROCEDURE

// ----- COST AND PENALTY MODEL -----

PROCEDURE EVALUATE_COST(ROUTE, T_MAX, T_BORDER, COST_PARAMS) RETURNS
TOTAL
    // DRONE OPERATING COSTS
    C_DRONE <- 0
    FOR EACH DRONE i USED IN ROUTE DO
        D_i <- TOTAL_DISTANCE(i, ROUTE) // km
        T_i <- TOTAL_FLIGHT_TIME(i, ROUTE) // hours
        N_CYCLES_i <- NUM_TAKEOFFS(i, ROUTE)
        C_DRONE <- C_DRONE
            + (COST_PARAMS.PER_CYCLE[i] * N_CYCLES_i)
            + (COST_PARAMS.PER_KM[i] * D_i)
            + (COST_PARAMS.PER_HOUR[i] * T_i)
    END FOR

    // OPERATOR COSTS
    T_TOTAL <- MAX_OVER_DRONES{ TOTAL_FLIGHT_TIME(i, ROUTE) } // mission
    wall time

```

Algorithm 1. *Cont.*

```

    N_STARTS <- TOTAL_TAKEOFFS(ROUTE)
    C_SALARY <- (T_TOTAL * COST_PARAMS.OP_HOURLY) + (N_STARTS *
COST_PARAMS.OP_PER_TAKEOFF)
    // PENALTIES
    PEN <- 0
    IF T_TOTAL > T_MAX THEN
        PEN <- (C_SALARY + C_DRONE) * POW(10, MIN(T_TOTAL - T_MAX, 10))
    ELSE IF T_TOTAL > T_BORDER THEN
        PEN <- (C_SALARY + C_DRONE) * (T_TOTAL - T_BORDER) / (T_MAX -
T_BORDER)
    END IF

    MISSED <- MISSED_WAYPOINTS(ROUTE)
    IF MISSED > 3 THEN
        PEN <- PEN + 1000000
    END IF

    TOTAL <- C_DRONE + C_SALARY + PEN
    RETURN TOTAL
END PROCEDURE

// ----- FAST PARTITION SCORING (STAGE I) -----

PROCEDURE QUICK_EVALUATE(PLAN, PART, ROAD, L, COST_PARAMS) RETURNS
COST_EST
    ROUTE_DRAFT <- DECODE(PLAN, PART, ROAD, L) // coarse simulation only
    COST_EST <- EVALUATE_COST_APPROX(ROUTE_DRAFT, COST_PARAMS)
    RETURN COST_EST
END PROCEDURE

```

Thus, the entire solution aligns with a two-stage scheme combining and implementing the proposed approach: first identifying the optimal polygon partitioning, then applying genetic optimization. All parameters used in the optimization process are additionally provided in Appendix A. Programming code can be found in Supplementary Materials section.

4. Results

To evaluate the algorithm, computational experiments were performed. For this, one of the fields was selected on which obstacles of different sizes were modeled. These obstacles model the presence of large structures or many small objects on the field, such as trees or equipment. Table 1 presents the calculation results. The columns “Baseline Approach” and “Partition Optimization” show the best total coverage cost over 25 iterations. Empirically, it was found that 10–15 iterations are usually sufficient for the algorithm to converge; the number 25 was chosen with a small margin. The first column shows results for the case without partitioning—routes are planned as if there are no obstacles, with dynamic adjustment for obstacle avoidance by contour following. The second column shows results for the proposed two-stage method. As can be seen, for three out of four fields, the proposed approach achieves a 20–30% advantage compared to the baseline approach.

Table 1. Calculation results.

Field	Baseline Approach	Partition Optimization	Improvement over the Baseline Approach (%)	Computation Time (s)	Number of Drones (Baseline/Partition Optimization)
Field 1—Large obstacle	465.300863	399.7649577	14.08	234	2/2
Field 2—Large elongated rectangular obstacle	375.7315059	331.2337709	11.84	190	1/1
Field 3—Two orthogonal rectangular obstacles	391.0776317	394.5266994	−0.88	178	3/2
Field 4—Multiple small obstacles	413.7058472	317.9067045	23.16	199	3/2

The “Computation Time (s)” column indicates the optimization runtime on an 8-core consumer-grade processor. It is important to note that the first stage of the algorithm—the preliminary search for the best partition—takes about 110–120 s. Hardware used for this experiment: CPU—Intel i7 1260P, RAM—32 Gb DDR4 3200 Mhz, SSD—Kioxia KXG60PNV2T04 XG6-P SSD—2 TB—M.2 2280—Internal—PCI Express NVMe—PCI Express NVMe 3.1 x4. Software environment: Windows 11, Python 3.9.13, shapely 2.0.6 for geometrical operations (intersection detection, area calculation, etc.), geopy 1.22.0, and vincenty 0.1.4 for geographical operations (coordinates transformation, distance evaluation, etc.), deap 1.3.1 as genetic algorithm framework, and podc 0.4.3 for polygon decomposition.

Figures 3–7 illustrate the experimental results for the specified fields with different obstacle configurations. In all figures red borders represent the field being traversed, yellow circles represent possible launch and landing locations, rectangular shapes and triangles represent obstacles, and colored lines represent drone routes. Figure 3 shows the routes for field number 1 in the baseline approach (upper row of figures) and in the case of the two-stage method (lower row of figures). The UAV routes are shown in different colors. The number of routes is equal to or greater than the number of UAVs.

In both cases, overflights for Field 1 can be performed by two drones. The UAVs take off at the point marked with a red circle in the figure and land there. The exception is the upper left figure, where the UAV lands at the point marked with a red circle. Figure 4 combines all routes for Field 1.

Figures 5–7 also shows the UAV flight routes for two planning options. The UAVs take off and land from one point in Fields 2 and 3. For Field 4, the route is planned so that when the mobile platform moves from left to right and from top to bottom (the route is indicated by yellow circles), the landing points of one drone become the takeoff points of another. In Figure 7, the colored circles indicate the takeoff points of the drones. The landing point is indicated by a black circle for all routes except the orange one, for which the takeoff point is orange, and landing point is green.

Appendices B–D show the UAV routes calculated using the two-stage method for Fields 2, 3, and 4, respectively.



Figure 3. Calculation results for Field 1. (**Top**) Tree routes in baseline approach, and (**bottom**) proposed two-stage method. The route of the mobile ground support platform is shown by yellow circles.



Figure 4. Combined UAV routes for Field 1. (**Left**) baseline approach, and (**right**) proposed two-stage method. The route of the mobile ground support platform is shown by yellow circles.



Figure 5. Calculation results for Field 2. (Left) baseline approach, and (right) proposed two-stage method.



Figure 6. Calculation results for Field 3. (Left) baseline approach, and (right) proposed two-stage method. In the right figure, all drone routes start and end at the same points (marked red circle), except the blue route, which takes off from one location and lands at another (green circle).

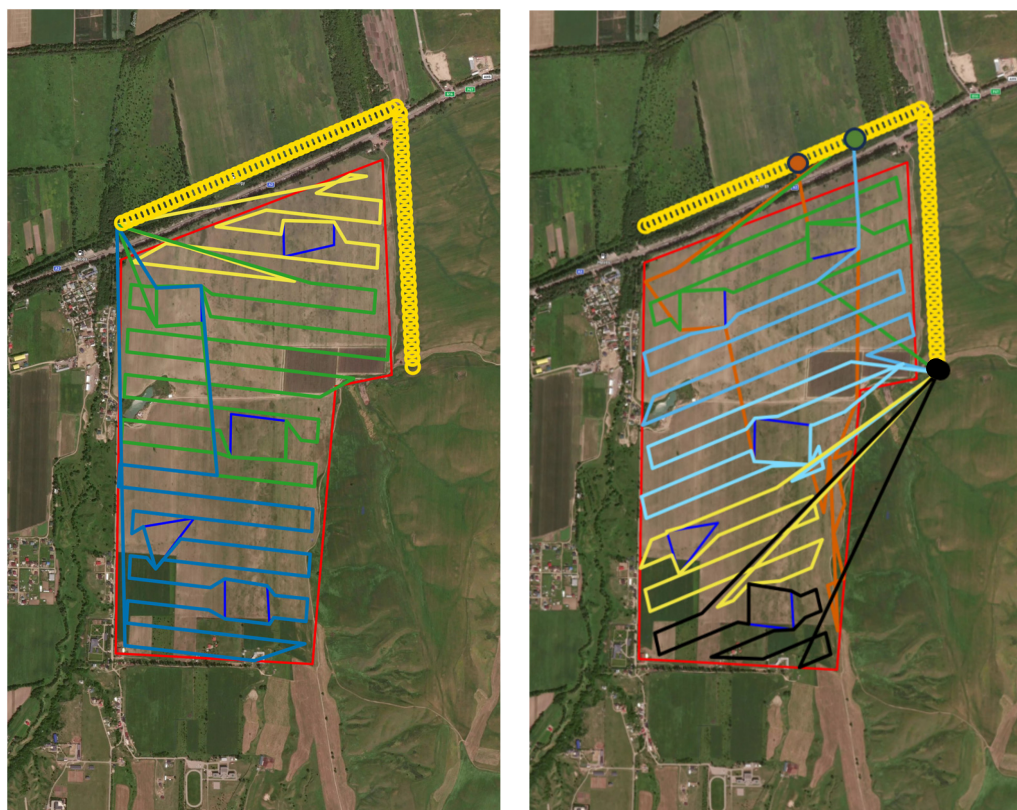


Figure 7. Calculation results for Field 4. (Left) baseline approach, and (right) proposed two-stage method.

5. Discussion of Results

Genetic algorithm-based optimization requires careful selection of parameters such as the number of iterations (generations), the population size, and the number of random requirements/individual combinations for the preliminary evaluation. The number of generations should be chosen so that the algorithm reaches stability (convergence), i.e., when further increases in the number of generations do not yield significant solution improvements. In practice, a good strategy is to analyze the dynamics of the objective function over intermediate iterations: if only minor improvements are observed over the last 10–20 generations (for example, less than 1–2%), convergence can be assumed. Refer to Figure 8 for convergence plots for several experiments: in most cases convergence is reached by the 10–15th generation.

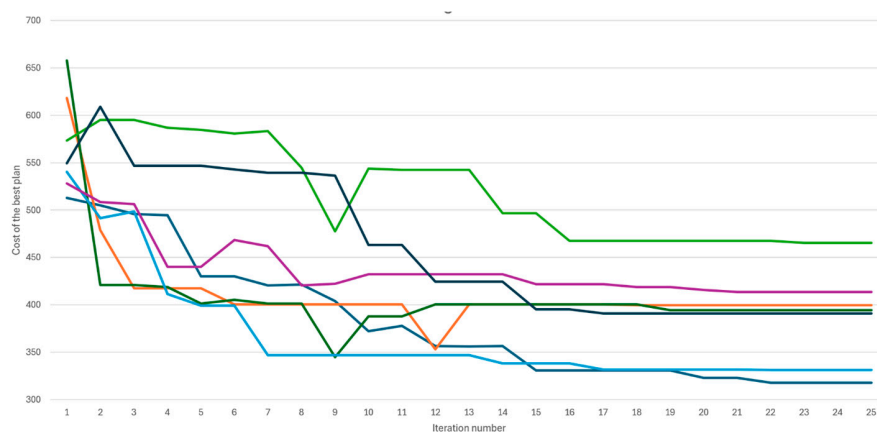


Figure 8. Convergence of cost of the best flight plan for each iteration. Each colored line represents convergence of cost of some given experiment.

Population size should provide sufficient diversity to avoid premature convergence to a local minimum. Typically, a population of 50 to 200 individuals is reasonable, but the optimal value depends on the problem's complexity and scale. The more complex the field geometry and the greater the number of drones, the larger the population should be to ensure solution diversity and effective algorithm performance.

The proposed genetic algorithm automatically selects which UAVs to deploy from an available fleet of heterogeneous drones, determining both the drone combination and their sequential order for surveying each sub-region. For instance, a single drone type might be used repeatedly across different sub-polygons, or various drones might alternate—such as using a faster but less maneuverable UAV for elongated regions, followed by a slower yet more agile UAV for geometrically complex areas. As illustrated by Table 1, optimal field subdivision typically reduces the diversity of drones required. This outcome arises because subdivision creates more uniform flight segments, diminishing the need to frequently switch UAV types. Conversely, the baseline approach often necessitates multiple drone types due to varying geometric complexities within a single contiguous route, resulting in inefficiencies.

The parameters NUM_RANDOM_INDIVS and NUM_RANDOM_REQUIREMENTS affect the quality of the partitioning evaluation stage. Increasing the number of random individuals and requirements improves the accuracy of optimal partition selection, but at the cost of increased computation time. In practice, it is advisable to start with a small value (e.g., 10–30), evaluate the stability and variability of the results, and then gradually increase these parameters if better solution quality is required.

The selection of UAV operational parameters (cost per kilometer, per takeoff, and per hour) should be based on real data regarding expenses and equipment depreciation. For example, the per-kilometer cost is calculated from the total maintenance and repair costs divided by the average drone mileage over its lifetime. The hourly cost reflects expenses for batteries, along with motor and electronics maintenance, as these are directly linked to flight time. The per-takeoff cost includes increased risks of mechanical damage and additional battery wear that occur during each takeoff–landing cycle.

Furthermore, when calculating the cost per takeoff, it is important to consider the risk of accidents and the need for equipment insurance. Each takeoff increases the likelihood of technical malfunctions and damage, making it reasonable to introduce a separate fee to cover such risks and associated expenses. Thus, correctly determining UAV operational parameters ensures the economic efficiency of planned operations and mitigates operational risks.

Due to the stochastic nature of the genetic algorithm, in some cases the solution found may be slightly inferior to the basic algorithm (see Table 1, scenario 3), but the average efficiency of the proposed algorithm is approximately 12 percent higher.

A combination of careful parameter tuning for the genetic algorithm and accurate estimation of UAV operational costs significantly improves the quality and economic efficiency of coverage routes, making the proposed solution a practical and reliable tool for precision agriculture applications.

6. Conclusions and Future Research Directions

This paper describes a route planning algorithm for surveying agricultural fields in the context of precision agriculture. The complexity of the problem arises from the following:

- the complex geometry of the field;
- the presence of internal obstacles;
- the heterogeneous composition of available UAVs;
- the ability (and desirability) to use a mobile ground platform.

A two-stage solution is proposed: (1) searching for an optimal fractional partition by rapidly iterating over several partitioning schemes and basic evaluation plans; (2) full genetic optimization of route parameters based on the chosen partitioning.

The key innovation compared to [50] is the two-stage optimization procedure: initially, a random set of field partitions is generated into several sub-polygons with pre-specified area proportions (considering internal obstacles). This approach can potentially provide more optimal solutions than the simple “Zamboni + obstacle avoidance” algorithm, depending on field shape, field size, and the size and number of obstacles. It avoids large suboptimal maneuvers in the overall trajectory and offers many degrees of freedom for economically efficient mission completion. It brings down the cost of the flights by 10–30% in most cases.

However, we should note some limitations of the study:

- (1) Genetic algorithms are metaheuristics based on stochastic selection methods that do not guarantee finding a global optimum.
- (2) The results of computational experiments have not yet been tested in real missions
- (3) The implemented software is its inability to handle non-convex obstacles, due to technical constraints in the geometric libraries used for transformations and intersection checks.

Promising directions for future research include the following:

- Incorporating detailed three-dimensional terrain and elevation variations (for example, when the drone can fly higher over tree belts, but at an increased energy cost).
- Enabling online route correction in response to changing conditions (weather factors, sudden drone failure).
- Investigating collision safety to ensure that the trajectories of different drones do not intersect at critically close distances during parallel operations.
- 3D simulation that takes into account the height of obstacles.
- Studying the influence of the initial partitioning of the field on the result of mission optimization.
- Investigating the problem of obstacle detection.
- Studying the influence of field relief on mission performance.
- Evaluation of simulation results during real missions.

One current limitation of the implemented software is its inability to handle non-convex obstacles due to technical constraints in the geometric libraries used for transformations and intersection checks.

Supplementary Materials: The following supporting information can be downloaded at <https://github.com/uavkz/SwarMown> (accessed on 1 August 2025).

Author Contributions: Conceptualization, K.Y. and R.I.M.; methodology, K.Y. and R.I.M.; software, K.Y.; validation, Y.K., E.M. and A.S.; investigation, K.Y., R.I.M. and A.S.; resources, R.I.M., A.S., E.M. and Y.K.; data curation, K.Y.; writing—original draft preparation, K.Y. and R.I.M.; writing—review and editing, R.I.M. and Y.K.; visualization, K.Y., E.M. and R.I.M.; supervision, Y.K.; project administration, K.Y.; funding acquisition, R.I.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (grants: № AP14869972 “Development and adaptation of computer vision and machine learning methods for solving precision agriculture problems using unmanned aerial systems”, № BR24992908 “Support system for agricultural crop production optimization via remote monitoring and artificial intelligence methods (Agroscope)”, and BR28713375 “Multipurpose Robotic UAV Platform for Remote Monitoring (AeroScope)”).

Data Availability Statement: The data presented in this study are openly available in <https://github.com/uavkz/SwarMown>.

Acknowledgments: The authors express their sincere gratitude to Victor Gopeenko for his active support of this work.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Parameters of the Genetic Algorithm

Input parameters

- Field polygon P ,
- Set of obstacles $\{H1, H2, \dots, Hs\}$, each $H_i \subset P$,
- Road (or roads) D , along which the ground platform moves,
- Set of drones $L = \{L1, \dots, L1\}$ with defined characteristics,
- p_i^{cycle} —fixed cost per takeoff–landing cycle for drone i (USD);
- p_i^{km} —cost per kilometer of flight for drone i (USD/km);
- p_i^{hour} —hourly operational cost for drone i (USD/h);
- h_{price} —operator’s hourly wage rate (USD);
- s_{price} —fixed operator payment per takeoff (USD).
- θ —flight angle ($0 \dots 360^\circ$);
- start_corner—one of («NW», «NE», «SW», «SE»);
- dronesList—ordered drone indices;
- carPointscarPointscarPoints—landing points along road (0 to 1 proportional positions);
- subPolygonsTraversalOrder—traversal order of sub-polygons r^* .

Selected parameters (parameters selected by stochastic search during the execution of the genetic algorithm)

- d_i —total distance traveled by drone i (km);
- t_i —total flight duration of drone i (hours).
- n_{starts} —total number of drone takeoffs;
- $T(x)$ —maximum flight duration among all drones (hours);
- θ —flight angle ($0 \dots 360^\circ$);
- start_corner—one of («NW», «NE», «SW», «SE»);
- dronesList—ordered drone indices;
- carPointscarPointscarPoints—landing points along road (0 to 1 proportional positions);
- subPolygonsTraversalOrder—traversal order of sub-polygons r^* .

Calculated parameters

- $C_{drone}(x)$ —total operational costs of the UAV fleet (USD);
- $C_{salary}(x)$ —total wages for drone operators (pilots) (USD);

Target value (the value to be minimized)

F_{total} —the objective function incorporated economic and time criteria

Appendix B

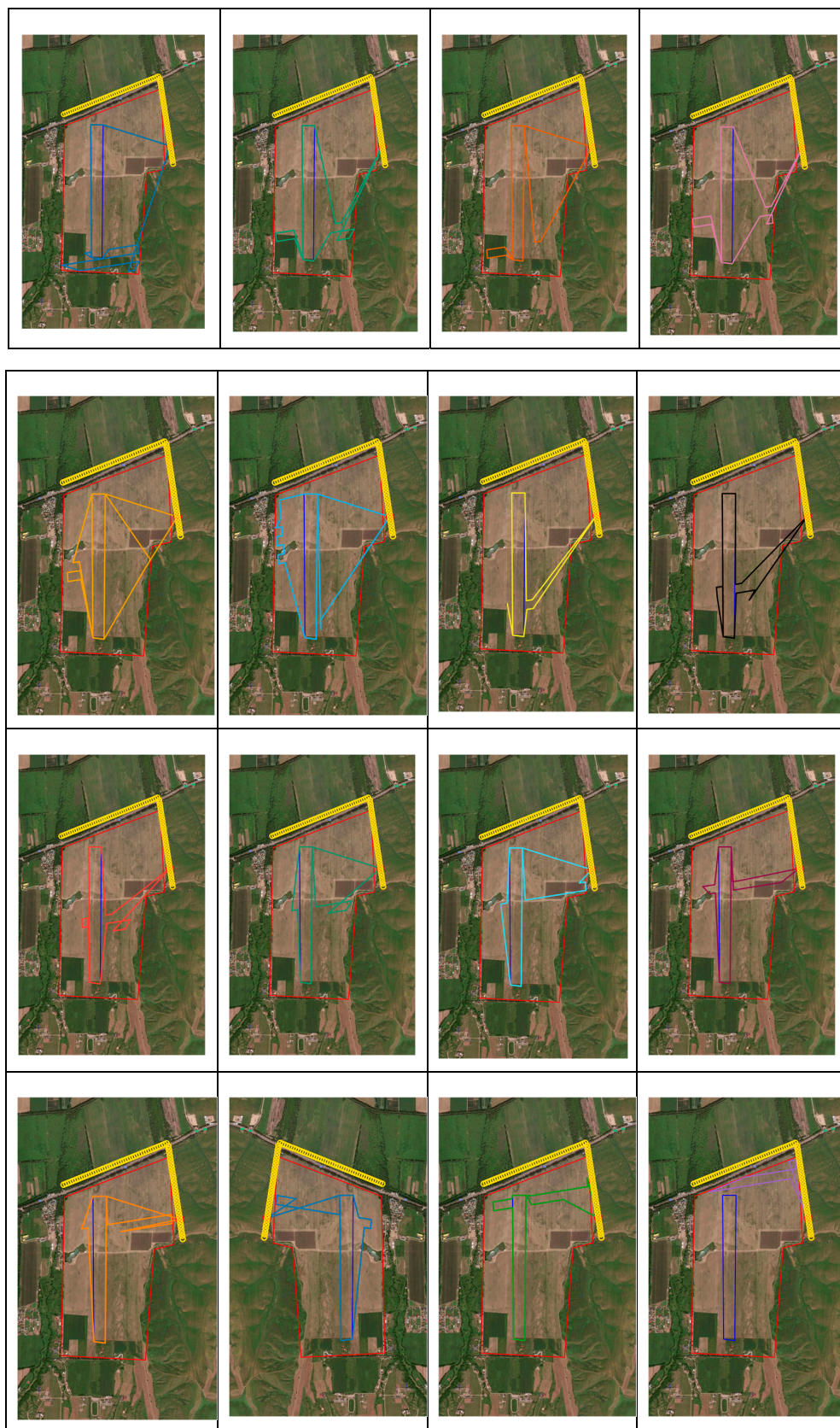


Figure A1. Calculation Results for Field 2 (Two-Stage Method).

Appendix C

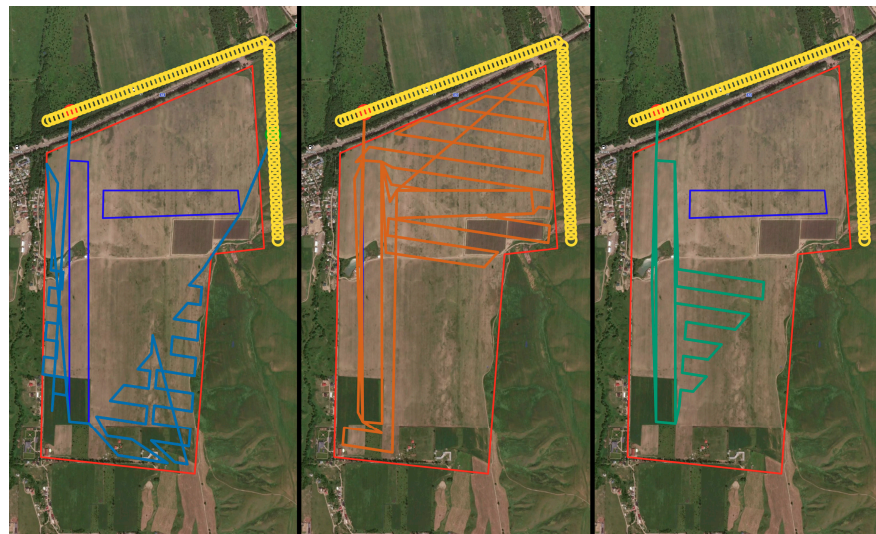


Figure A2. Calculation Results for Field 3 (Two-Stage Method).

Appendix D



Figure A3. Calculation Results for Field 4 (Two-Stage Method).

References

1. Mogili, U.R.; Deepak, B.B.V.L. Review on Application of Drone Systems in Precision Agriculture. *Procedia Comput. Sci.* **2018**, *133*, 502–509. [CrossRef]
2. Park, S.; Choi, Y. Applications of Unmanned Aerial Vehicles in Mining from Exploration to Reclamation: A Review. *Minerals* **2020**, *10*, 663. [CrossRef]
3. Gailler, L.; Labazuy, P.; Régis, E.; Bontemps, M.; Souriot, T.; Bacques, G.; Carton, B. Validation of a New UAV Magnetic Prospecting Tool for Volcano Monitoring and Geohazard Assessment. *Remote Sens.* **2021**, *13*, 894. [CrossRef]
4. Erdelj, M.; Król, M.; Natalizio, E. Wireless Sensor Networks and Multi-UAV Systems for Natural Disaster Management. *Comput. Netw.* **2017**, *124*, 72–86. [CrossRef]
5. Mukhamediev, R.; Alexandrov, D.A.; Boukhanovsky, A.V.; Chugunov, A.V.; Kabanov, Y.; Koltsova, O.; Musabirov, I. Intelligent Unmanned Aerial Vehicle Technology in Urban Environments. In *Digital Transformation and Global Society; Communications in Computer and Information Science*; Springer: Cham, Switzerland, 2020; Volume 1242, pp. 312–325. [CrossRef]
6. Hu, J.; Niu, H.; Carrasco, J.; Lennox, B.; Arvin, F. Fault-Tolerant Cooperative Navigation of Networked UAV Swarms for Forest Fire Monitoring. *Aerosp. Sci. Technol.* **2022**, *123*, 107494. [CrossRef]
7. Hodgson, J.C.; Baylis, S.M.; Mott, R.; Herrod, A.; Clarke, R.H. Precision Wildlife Monitoring Using Unmanned Aerial Vehicles. *Sci. Rep.* **2016**, *6*, 22574. [CrossRef]
8. Yuan, S.; Li, Y.; Bao, F.; Xu, H.; Yang, Y.; Yan, Q.; Lin, J. Marine Environmental Monitoring with Unmanned Vehicle Platforms: Present Applications and Future Prospects. *Sci. Total Environ.* **2023**, *858*, 159741. [CrossRef]
9. Medvedev, A.; Telnova, N.; Alekseenko, N.; Koshkarev, A.; Kuznetchenko, P.; Asmaryan, S.; Narykov, A. UAV-Derived Data Application for Environmental Monitoring of the Coastal Area of Lake Sevan, Armenia with a Changing Water Level. *Remote Sens.* **2020**, *12*, 3821. [CrossRef]
10. Jenssen, R.; Roverso, D. Intelligent Monitoring and Inspection of Power Line Components Powered by UAVs and Deep Learning. *IEEE Power Energy Technol. Syst. J.* **2019**, *6*, 11–21. [CrossRef]
11. Jordan, S.; Moore, J.; Hovet, S.; Box, J.; Perry, J.; Kirsche, K.; Tse, Z.T.H. State-of-the-Art Technologies for UAV Inspections. *IET Radar Sonar Navig.* **2018**, *12*, 151–164. [CrossRef]
12. Ham, Y.; Han, K.K.; Lin, J.J.; Golparvar-Fard, M. Visual Monitoring of Civil Infrastructure Systems via Camera-Equipped Unmanned Aerial Vehicles (UAVs): A Review of Related Works. *Vis. Eng.* **2016**, *4*, 1–8. [CrossRef]
13. Khan, N.A.; Jhanjhi, N.Z.; Brohi, S.N.; Usmani, R.S.A.; Nayyar, A. Smart Traffic Monitoring System Using Unmanned Aerial Vehicles (UAVs). *Comput. Commun.* **2020**, *157*, 434–443. [CrossRef]
14. Drone Inspections Help Cut Pollution by Half. Available online: <https://gulfnews.com/uae/environment/drone-inspections-help-cut-pollution-by-half-1.2263928> (accessed on 16 September 2021).
15. Mohsan, S.A.H.; Othman, N.Q.H.; Li, Y.; Alsharif, M.H.; Khan, M.A. Unmanned Aerial Vehicles (UAVs): Practical Aspects, Applications, Open Challenges, Security Issues, and Future Trends. *Intell. Serv. Robot.* **2023**, *16*, 109–137. [CrossRef] [PubMed]
16. Telli, K.; Kraa, O.; Himeur, Y.; Ouamane, A.; Boumehraz, M.; Atalla, S.; Mansoor, W. A Comprehensive Review of Recent Research Trends on Unmanned Aerial Vehicles (UAVs). *Systems* **2023**, *11*, 400. [CrossRef]
17. Erkec, T.Y.; Hajiyev, C. Swarm Architecture of UAVs. In *Progress in Sustainable Aviation*; Karakoc, T.H., Colpan, C.O., Dalkiran, A., Eds.; Sustainable Aviation; Springer: Cham, Switzerland, 2022; pp. 15–34. [CrossRef]
18. Mohsan, S.A.H.; Khan, M.A.; Noor, F.; Ullah, I.; Alsharif, M.H. Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive Review. *Drones* **2022**, *6*, 147. [CrossRef]
19. Shakhathreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* **2019**, *7*, 48572–48634. [CrossRef]
20. On the Approval of the Rules for the Use of Unmanned Aerial Systems in the Airspace of the Republic of Kazakhstan. Available online: <https://adilet.zan.kz/rus/docs/V2000022031> (accessed on 15 August 2025).
21. Mukhamediev, R.I.; Symagulov, A.; Kuchin, Y.; Zaitseva, E.; Bekbotayeva, A.; Yakunin, K.; Assanov, I.; Levashenko, V.; Popova, Y.; Akzhalova, A.; et al. Review of Some Applications of Unmanned Aerial Vehicles Technology in the Resource-Rich Country. *Appl. Sci.* **2021**, *11*, 10171. [CrossRef]
22. Wyngaard, J.; Barbieri, L.; Thomer, A.; Adams, J.; Sullivan, D.; Crosby, C.; Parr, C.; Klum, J.; Shrestha, S.R.; Bell, T. Emergent Challenges for Science sUAS Data Management: Fairness through Community Engagement and Best Practices Development. *Remote Sens.* **2019**, *11*, 1797. [CrossRef]
23. Haghray, A.A.; Ghaemi, S.; Badamchizadeh, M.A. Modeling and Solving Multi-Objective Path Planning Problem for Cooperative Cable-Suspended Load Transportation Considering the Time Variable Risk. *IEEE Access* **2025**, *13*, 11704–11719. [CrossRef]
24. Ccari, L.F.C.; Yanyachi, P.R.; Luque, J.C.C.; Yanyachi, D. Distributed Robust Adaptive Control for Finite-Time Flight Formation of Multi-Quadcopter Systems with Large Lumped Uncertainties. *IEEE Access* **2024**, *12*, 113384–113405. [CrossRef]

25. Bao, X.; Hu, Z.H.; Huang, Y. Routing a Fleet of Drones from a Base Station for Emission Detection of Moving Ships by Genetic Algorithm. *J. Mar. Sci. Eng.* **2024**, *12*, 891. [CrossRef]
26. Alqudsi, Y.; Makaraci, M. UAV Swarms: Research, Challenges, and Future Directions. *J. Eng. Appl. Sci.* **2025**, *72*, 12. Available online: <https://jeas.springeropen.com/articles/10.1186/s44147-025-00582-3> (accessed on 15 August 2025). [CrossRef]
27. Muslimov, T.Z. Algorithms of Autonomous Fixed-Wing UAVs Formation Control via Vector Field Method. *Syst. Control Commun. Secur.* **2019**, *4*, 187–214. (In Russian)
28. Huang, G.; Hu, M.; Yang, X.; Huang, F. Multi-UAV Cooperative Mission Assignment Based on Hybrid WAFC-RRAS Algorithm. *IEEE Access* **2023**, *11*, 98132–98152. [CrossRef]
29. Martinez-Alpiste, I.; Golcarenenji, G.; Wang, Q.; Alcaraz-Calero, J.M. Search and Rescue Operation Using UAVs: A Case Study. *Expert Syst. Appl.* **2021**, *178*, 114937. [CrossRef]
30. Fang, Z.; Savkin, A.V. Strategies for Optimized UAV Surveillance in Various Tasks and Scenarios: A Review. *Drones* **2024**, *8*, 193. [CrossRef]
31. Ahuraka, F.; Mcnamee, P.; Wang, Q.; Ahmadabadi, Z.N.; Hudack, J. Chaotic Motion Planning for Mobile Robots: Progress, Challenges, and Opportunities. *IEEE Access* **2023**, *11*, 134917–134939. [CrossRef]
32. Kumar, K.; Kumar, N. Region Coverage-Aware Path Planning for Unmanned Aerial Vehicles: A Systematic Review. *Phys. Commun.* **2023**, *59*, 102073. [CrossRef]
33. QGroundControl. Available online: <http://qgroundcontrol.com/> (accessed on 7 November 2024).
34. Dronekit. Available online: <https://dronekit.io/> (accessed on 7 November 2024).
35. Ardupilot. Available online: <https://ardupilot.org/planner/docs/common-loading-firmware-onto-pixhawk.html> (accessed on 7 November 2024).
36. UgCS. Available online: <https://www.sphengineering.com/flight-planning/ugcs> (accessed on 7 November 2024).
37. Vélez, S.; Mier, G.; Ariza-Sentís, M.; Valente, J. Integrated Framework for Multipurpose UAV Path Planning in Hedgerow Systems Considering the Biophysical Environment. *Crop Prot.* **2025**, *187*, 106992. [CrossRef]
38. Wang, M.; Zhang, D.; Li, C.; Zhang, Z. Multiple Fixed-Wing UAVs Collaborative Coverage 3D Path Planning Method for Complex Areas. *Def. Technol.* **2025**, *47*, 197–215. [CrossRef]
39. Maaji, S.S.; Landa-Silva, D. UAV Path Planning for Area Coverage and Energy Consumption in Oil and Gas Exploration Environment. In *Computational Logistics, Proceedings of the International Conference on Computational Logistics, Berlin, Germany, 6–8 September, 2023*; Springer: Cham, Switzerland, 2023; pp. 467–481.
40. Zhang, L.; Su, J.; Xu, C.; He, T. Coverage Path Planning with Minimum Cost Suitable for Sidewalk. In *Proceedings of the 2023 3rd International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)*, Wuhan, China, 15–17 December 2023; pp. 801–805.
41. Li, J.; Ren, W. A Multi-Machine Cooperation Full Coverage Path Planning Method in Agriculture. In *Proceedings of the 2023 4th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, Nanjing, China, 16–18 June 2023; pp. 713–716.
42. Li, J.; Sheng, H.; Zhang, J.; Zhang, H. Coverage Path Planning Method for Agricultural Spraying UAV in Arbitrary Polygon Area. *Aerospace* **2023**, *10*, 755. [CrossRef]
43. Plessen, M. Path Planning for Spot Spraying with UAVs Combining TSP and Area Coverages. *arXiv* **2024**, arXiv:2408.08001. [CrossRef]
44. Bine, L.M.S.; Boukerche, A.; Ruiz, L.B.; Loureiro, A.A.F. A Novel Ant Colony-Inspired Coverage Path Planning for Internet of Drones. *Comput. Netw.* **2023**, *235*, 109963. [CrossRef]
45. Karthik, K.; Balasubramanian, C. Improved Green Anaconda Optimization Algorithm-Based Coverage Path Planning Mechanism for Heterogeneous Unmanned Aerial Vehicles. *Sustain. Comput.* **2024**, *42*, 100961. [CrossRef]
46. Aggarwal, S.; Kumar, N. Path Planning Techniques for Unmanned Aerial Vehicles: A Review Solutions and Challenges. *Comput. Commun.* **2020**, *149*, 270–299. [CrossRef]
47. Ahmadi, S.M.; Kebriaei, H.; Moradi, H. Constrained Coverage Path Planning: Evolutionary and Classical Approaches. *Robotica* **2018**, *36*, 904–924. [CrossRef]
48. Önlér, E. Comparative Analysis of Genetic and Greedy Algorithm for Optimal Drone Flight Route Planning in Agriculture. *Anadolu Tarım Bilim. Derg.* **2024**, *39*, 129–142. [CrossRef]
49. Barzola, P.D.; Baños, E.H.; Baños, L.H.; Moreira, W.M.; Muentes, E.H. Review of Genetic Algorithms Applied to Agriculture. In *Proceedings of the 2024 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, Sydney, Australia, 25–27 July 2024; pp. 1–8.
50. Mukhamediev, R.I.; Yakunin, K.; Aubakirov, M.; Assanov, I.; Kuchin, Y.; Symagulov, A.; Levashenko, V.; Zaitseva, E.; Sokolov, D.; Amirgaliyev, Y. Coverage Path Planning Optimization of Heterogeneous UAVs Group for Precision Agriculture. *IEEE Access* **2023**, *11*, 5789–5803. [CrossRef]

51. Wen, J.; Wang, F.; Su, Y. A Bi-Layer Collaborative Planning Framework for Multi-UAV Delivery Tasks in Multi-Depot Urban Logistics. *Drones* **2025**, *9*, 512. [[CrossRef](#)]
52. Zhang, Y.; Xu, W.; Ye, H.; Shi, Z. A Two-Stage Optimization Framework for UAV Fleet Sizing and Task Allocation in Emergency Logistics Using the GWO and CBBA. *Drones* **2025**, *9*, 501. [[CrossRef](#)]
53. Pode Library. Available online: <https://github.com/GeorgySk/pode> (accessed on 7 November 2024).
54. Hert, S.; Lumelsky, V. Polygon Area Decomposition for Multiple-Robot Workspace Division. *Int. J. Comput. Geom. Appl.* **1998**, *8*, 437–466. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.