The Aeronautical Journal (2025), **129**, pp. 2827–2842
doi:10.1017/aer.2025.10030

ROYAL AERONAUTICAL SOCIETY

RESEARCH ARTICLE

Optimal flight path planning for micro-UAV-based aerial imaging using the colony-based search algorithm

P. Civicioglu

Department of Aircraft Electrics and Electronics, Faculty of Aeronautics and Astronautics, Erciyes University, Kayseri, Türkiye
Email: civici@erciyes.edu.tr

Received: 8 April 2025; **Revised:** 7 May 2025; **Accepted:** 7 May 2025

Keywords: micro-UAV; flight mission planning; colony-search algorithm; ground-sample distance

Abstract

Micro-UAV systems used for metric purposes are highly capable of capturing relatively high-resolution, chromatically stable aerial images at low altitudes. In micro-UAV-based aerial imaging-based structure-from-motion (*a-SfM*) applications, the flight mission planning problem can be customised to achieve different objectives. The requirement for minimising the time spent in the air, which is crucial for energy conservation, can be achieved by designing the shortest possible flight path. Spatial resolution in the captured aerial images can be significantly preserved by maintaining the ground sampling distance (GSD) value within a 95% confidence interval throughout the flight path. Fuel efficiency can be improved by minimising the number of turning manoeuvers required to follow the flight path during the flying mission. In this paper, four distinct flight mission planning processes are delineated to enable the energy-efficient and effective implementation of aerial imaging missions, with their associated parameters optimised using the colony-based search algorithm (CSA). The obtained experimental results demonstrate that the proposed flight mission planning processes are highly successful in the energy-efficient and effective execution of aerial imaging missions.

Nomenclature

Below are the technical expressions for some of the variables used in Fig. 2

GSD	ground sample distance
Vertices	ordered vertex coordinates of the closed polygon ([$N \times 2$] as [x; y])
p_{Endlap}	forward-overlap ratio ($0 < p < 1$)
$q_{Sidelap}$	side-overlap ratio $0 < q < 1$
H	Flight altitude (meters)
ImageWidth	image width in pixels
ImageHeight	image height in pixels
f	focal length of camera as meter
ccdWidth	width of imaging sensor as meter
azimuth	direction angle for the flight start (in degrees)
flightPaths	cell array containing x and y spatial coordinates for each flight path
waypoints	combined waypoints of all flight paths ($M \times 2$ sized matrix)
totalTurns	total number of turns made by the drone
dp	vertices of turn locations

The following are the technical expressions for some of the variables used in Fig. 3.

$U\left(low_j, up_j\right)$	continuous uniform distribution with parameters a and b
$U\{1, 2, 3\}$	discrete uniform distribution with parameters a and b
ObjFun	objective function

© The Author(s), 2025. Published by Cambridge University Press on behalf of Royal Aeronautical Society.

https://doi.org/10.1017/aer.2025.10030 Published online by Cambridge University Press

rperm (a)	randomly shuffles the positions of integer values between [1, a] according to a uniform distribution
shuffle (a, b)	randomly shuffles integers $[1, a]$, returns first b values
rperm $(a; b)$	randomly shuffles the positions of integer values between [1, a] according to a uniform distribution and returns the first b of them
rnd ()	generates a random real-valued number between [0,1] according to the rules of a uniform distribution each time it is called
rnd $(a;b)$	generates a 2D matrix with 'a' rows and 'b' columns each time it is called; he elements of the matrix are random values within the range [0,1], following the rules of uniform distribution
sign $(\cdot)$	returns sign of numeric value $(\cdot)$
$\oslash$	element-wise division operator (Hadamard division operator)
sign $(\cdot)$	returns the sign of the numeric value (.)
$\circ$	element-wise multiplication operator (Hadamard multiplication operator)
LevyDist $(\alpha, \beta, N, 1)$	generates a matrix with uniform random-valued elements having N rows and 1 column, using a Levy distribution with parameters (α, β)
zeros (a, b)	generates a 2D matrix of size [a b] with all elements set to zero
$\lceil \cdot \rceil$	In discrete mathematics, represents the ceiling function
sortindex $(a, 'ascend')$	sorts the elements of the vector 'a' in ascending order and returns the resulting sorted vector
rndi $([c, d])$	generates a uniform single integer value between [c d]
abs $(\cdot)$	absolute value function

1.0 Introduction

An optimally designed flight mission planning (FMP) process [1–3] reduces the energy and cost required for the associated flight operations while indirectly enhancing the accuracy of applications based on advanced structure-from-motion (*a-SfM*) techniques, such as neural radiance fields (NeRF), Gaussian splatting, multi-view stereo (MVS) and structure from motion (SfM). Therefore, FMP is a critical process that ensures the effective capture of aerial images, enabling accurate three-dimensional reconstructions and geospatial analyses [4–6].

FMP involves designing a detailed strategy that specifies parameters such as altitude, speed, trajectory, camera settings and image timing to meet specific project goals. The resulting data must align with rigorous standards of accuracy, coverage and usability for applications like topographic mapping, 3D modelling and terrain analysis. Beyond technical precision, the flight plan also aims to optimise operational efficiency, minimise costs and reduce environmental impacts [7–13].

The first step in FMP is to understand the project's objectives. For instance, creating a digital elevation model [14–16], an orthophoto, or a detailed 3D reconstruction requires distinct considerations. Key variables include the number and placement of ground control points (GCPs), which are essential for georeferencing and validating the accuracy of the final outputs. GCP distribution [17, 18] plays a pivotal role in ensuring reliable results. Additionally, constraints such as terrain complexity, weather conditions and equipment limitations must be factored into the plan. A well-designed trajectory ensures comprehensive coverage while adhering to these constraints, directly influencing data quality and processing efficiency [19–23].

Fuel efficiency is a growing concern in *a-SfM* [17, 24, 25] due to rising operational costs and the need for sustainable practices. Optimising flight paths can significantly reduce fuel consumption by minimising unnecessary manoeuvres and excessive distances travelled. For instance, adopting grid-based flight patterns with minimal turns and maintaining consistent altitudes optimised for the camera's focal length and the desired ground sample distance (GSD) significantly improves fuel efficiency. Such refinements in flight planning not only reduce carbon emissions but also enhance the economic feasibility of large-scale aerial surveys. Achieving a balance between thorough coverage and efficient operations is crucial for the success of modern FMP applications [26–32].

https://doi.org/10.1017/aer.2025.10030 Published online by Cambridge University Press

SfM-based aerial mapping is a state-of-the-art method for aerial imaging. To compute high-accuracy spatial coordinates of points on a topographic surface using SfM-based aerial imaging, completing the aerial triangulation process with high precision is essential. High-accuracy aerial triangulation aids in tightly linking sequential stereo models. Achieving high-quality aerial triangulation ensures the production of high-quality topographic surface data. The quality of the aerial triangulation process heavily depends on obtaining highly accurate camera projection centre coordinates. However, due to the dynamic nature of the aerial imaging process, camera projection centre coordinates alone may not suffice to achieve the required accuracy for aerial triangulation. Therefore, GCPs with sufficient spatial distribution are established across the imaged topographic surface. Accuracy of aerial triangulation is the highest at GCP locations but oscillates in a parabolic form between GCPs. The amplitude of these oscillations varies depending on the distances between GCPs and on whether the GCPs include (x, y, z) information or only (z).

The number of GCPs can be defined as a function of the desired accuracy of the point cloud resulting from SfM-based aerial imaging. Studies using a DJI Phantom 4 Pro drone have shown that a GCP network spaced 125–150 m apart on the topographic surface is sufficient to achieve a point cloud with better than 5 cm spatial accuracy in SfM-based processing. Real-time kinematic (RTK)-enabled drones typically utilise a local correction service that supports the networked transport of RTCM via internet protocol (NTRIP).

Accuracy is a critical factor in FMP [28, 33–35], as the primary objective is to generate data that accurately reflects the physical environment [36, 37]. Key elements such as flight altitude, camera calibration [38] and imaging geometry must be meticulously calculated to achieve this goal. Altitude plays a significant role in determining resolution–lower altitudes produce higher levels of detail but cover smaller areas, whereas higher altitudes offer broader coverage but with reduced precision. Consistent imaging geometry is essential, particularly for processes like bundle adjustment in 3D reconstruction, as it ensures reliable and accurate results. By optimising flight parameters, planners can reduce errors and ensure that the derived coordinates closely match real-world measurements.

Forward-overlap (*Endlap*) and side-overlap (*Sidelap*) ratios of captured adjacent aerial images [39, 40] are critical to FMP, directly influencing 3D scene reconstruction using *a-SfM* methods. Typically, 60–80% *endlap* along the flight path and 40–70% *sidelap* between adjacent paths are required to ensure sufficient common features for feature matching and triangulation. Inadequate overlaps may cause coverage gaps or insufficient data for stereo processing, leading to incomplete or inaccurate models. These ratios vary based on camera field of view, terrain complexity and application needs, making them essential to flight planning. Epipolar geometry principles enhance planning by ensuring consistent imaging configurations, aiding relative orientation and stereo matching while minimising distortions for reliable 3D reconstructions. Flight planning balances precision, efficiency and sustainability by optimising paths, maintaining accuracy and adhering to geometric principles, yielding high-quality data for geospatial analysis and cost-effective, eco-friendly mapping.

Micro-UAV path planning problems predominantly involve discrete variables, exhibit complex structures, and are often non-differentiable. Consequently, evolutionary computing methods are frequently employed [3, 30, 41–50] to address micro-UAV path planning problems [51–54]. In this paper, four distinct problems defined for micro-UAVs are solved using CSA [51], a swarm-based evolutionary computing method. CSA is specifically designed to compute the global minimum of real-valued numerical problems and it operates on a multi-swarm basis and incorporates a structurally unique artificial morphogenesis process.

The contributions of the proposed method are as follows:

- The FMP problem was heuristically solved using CSA across four distinct scenarios, each involving varying flight and terrain coverage conditions.
- The FMP problem was solved to minimise the total flight distance at a fixed flying altitude.
- The FMP problem was solved to minimise 95% of the GSD values of pixels in captured aerial images along the flight path.

https://doi.org/10.1017/aer.2025.10030 Published online by Cambridge University Press

- The FMP problem was solved to minimise 95% of the GSD values of pixels in captured aerial images while reducing the number of turns along the flight path.
- The FMP problem was solved to maximise the number of aerial stereo images captured, ensuring complete coverage of the given terrain.

The rest of this paper is organised as follows: In Section 2, flight path planning for aerial imaging is introduced. In Section 3, the CSA is presented. In Section 4, experiments are discussed, and finally, conclusions are provided in Section 5.

2.0 Flight path planning for aerial imaging

Aerial imaging-based topographic mapping methods can only capture aerial images that meet specific geometric requirements through the use of an optimised FMP process. Therefore, FMP is an essential process for aerial imaging-based topographic mapping applications. From the perspective of aerial imaging-based mapping methods, FMP ensures that the captured aerial images achieve the desired numerical overlap limits in both forward and side directions. This process facilitates the mutual orientation of consecutive images and enables the generation of a 3D stereo model of the imaged area. To meet the need for high-accuracy aerial mapping production, it is crucial to pre-plan an adequate GSD [55–58] value during the initial step of FMP. In the experiments conducted in this paper, TUSAGA-AKTIF was used for the global positioning of the micro-UAV. TUSAGA-AKTIF is a GPS-based real-time global positioning network [59].

Variations in elevation on the topographic surface imaged using micro-UAVs can cause numerical fluctuations in the GSD. Additionally, micro-UAV platforms generally have very limited onboard energy resources. For this reason, optimising the flight plan to minimise fluctuations in the GSD is a necessity for efficient use of the micro-UAV's onboard energy. The general analytical definition of GSD is provided in Equation (1).

$$\mathrm{GSD}_{\mathrm{meter/pixel}} = \frac{H_{meter} \cdot ccdWidth_{meter/pixel}}{f_{meter}} = \frac{H_{meter} \cdot SensorWidth_{mm}}{f_{mm} \cdot ImageWidth_{pixels}}, \tag{1}$$

where, H represents the flight altitude (i.e. elevation hight), $SensorWidth$ denotes the physical size of the sensor in millimetres, and f corresponds to the focal length. $ImageWidth$ and $ccdWidth_{meter/pixel}$ denote the width of the image, measured in pixels, and the width of a single pixel in the camera's CCD, measured in meters, respectively. In this paper, altitude measurements were obtained using barometric methods.

The height of a single image footprint corresponds to the flight direction of the micro-UAV. The width of a single image footprint on the ground ($GSDW_{\mathrm{meter}}$) and the height of a single image footprint on the ground ($GSDH_{\mathrm{meter}}$) are calculated using Equations (2) and (3), respectively.

$$GSDW_{\mathrm{meter}} = GSD_{meter/pixels} \cdot ImageWidth_{pixels}, \tag{2}$$

$$GSDH_{\mathrm{meter}} = GSD_{meter/pixels} \cdot ImageHeigh_{pixels}. \tag{3}$$

Aerial imaging based mapping methods necessitates a sufficient degree of image overlap to ensure accurate processing and reconstruction. Specifically, forward overlap, which refers to the overlapping area along the flight path, is typically maintained within a range of 60% to 80%. Similarly, side overlap, representing the overlapping region between adjacent flight lines, generally falls within the range of 40% to 70%. These overlap parameters are critical for achieving high-quality results in photogrammetric analyses.

The distance between adjacent images, $Endlap_{meter}$, and distance between side-flight lines, $Sidelap_{meter}$, are calculated by using Equations (4) and (5):

$$Endlap_{meter} = GSDW_{meter} \cdot \left(1 - p_{Endlap}\right), \tag{4}$$

$$Sidelap_{meter} = GSDH_{meter} \cdot \left(1 - q_{Sidelap}\right), \tag{5}$$

https://doi.org/10.1017/aer.2025.10030 Published online by Cambridge University Press

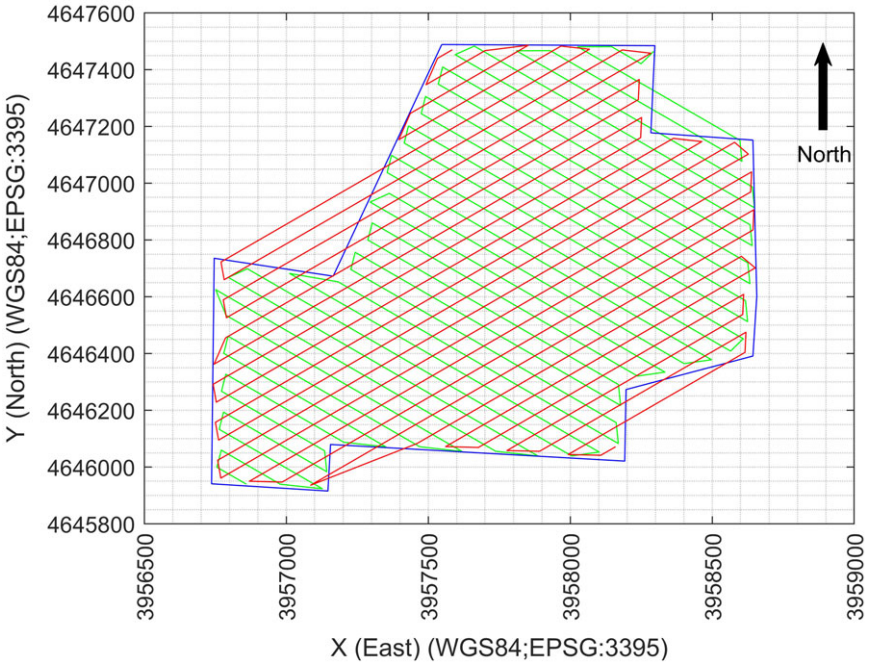


Figure 1. *Blue-coloured convex polygon is used for FMP optimisation tests. The spatial coordinates are defined in the datum of WGS84 and projection system of EPSG:3395 (WGS84/World Mercator). The red-coloured FMP is associated with an azimuth* $= 30^0$*, whereas the green-coloured FMP corresponds to an azimuth* $= -30^0$*.*

where p_{Endlap} and $q_{Sidelap}$ denote forward-overlap and side-overlap values of aerial images as %, respectively.

In this paper, the 2D vertex coordinates of the polygon for which the relevant flight plan is to be created have been defined using the WGS84 datum and the EPSG:3395 (WGS84/World Mercator) projection system. The *Endlap* and *Sidelap* values were used to completely cover the 2D polygon surface, where the corresponding FMP is to be performed, with a point-mesh grid of size $\left[\text{Endlap}\ \ \text{Sidelap}\right]$. The central position of the generated point-mesh coincides with the centre of the corresponding 2D polygon. Each point in the point-mesh represents a waypoint. Each waypoint corresponds to an aerial image capture location. By optimising the *azimuth* of the Point-Mesh relative to the centre of the 2D polygon, the number of waypoints remaining inside the polygon is maximised, thereby achieving maximum coverage that satisfies the flight parameters over the corresponding 2D polygon. In this paper, the relevant *azimuth* value is calculated using an evolutionary optimisation algorithm called the colony-based search algorithm (CSA) [51]. The waypoints are organised as an interlacing-path to generate the flight path. The number of turns on the interlacing-path corresponds to the total number of turns.

An FMP with an optimal flight path and conditions designed for a micro-UAV facilitates the efficient use of on-board energy. In Fig. 1, two different non-optimal FMPs, calculated for $azimuth = 30^0$ and $azimuth = -30^0$, are shown in red and green, respectively, over the test polygon used in the experiments presented in this paper. The additional flying mission parameters for these flights are provided as follows: $p_{Endlap} = 0.80$, $q_{Sidelap} = 0.70$, $H = 200$ m, $ImageWidth = 5472$ pixels, $ImageHeight = 3648$ pixels, $f = 8.8 \cdot 10^{-3}$ m, and $ccdWidth = 2.41 \cdot 10^{-6}$ m. For an FMP that ensures optimal conditions, the *azimuth* angle should be optimised according to the objective function.

The pseudo-code of the FMP method proposed in this paper is provided in Fig. 2, where the lines #1–4 compactly express the calculation processes related to the GSD value. The process for calculating

https://doi.org/10.1017/aer.2025.10030 Published online by Cambridge University Press

Input :

- $polygonVertices$:Coordinates of the polygon vertices ($N \times 2$ matrix)
- p: Forward overlap rate ($0 < p < 1$)
- b: Lateral overlap rate ($0 < b < 1$)
- h: Flight altitude (meters)
- t: azimuth angle (in radians)

Output:

- $flightPaths$: Coordinates for each flight path (cell array)
- $waypoints$: All waypoints of flight paths ($M \times 2$ matrix)
- $totalTurns$: Total number of turns made by the drone

1 $cameraFocalLength \leftarrow 35 \times 10^{-3}$ ▷ `Focal length`
2 $sensorWidth \leftarrow 36 \times 10^{-3}$ ▷ `Camera sensor width`
3 $sensorHeight \leftarrow 24 \times 10^{-3}$ ▷ `Camera sensor height`
4 $GSD \leftarrow \frac{sensorWidth \cdot h}{cameraFocalLength}$ ▷ `GSD`
5 $EndLap \leftarrow GSD \cdot (1 - p)$ ▷ `Forward overlap`
6 $Sidelap \leftarrow GSD \cdot (1 - b)$ ▷ `Lateral overlap`
7 $centroid \leftarrow \text{mean}(polygonVertices, 1)$ ▷ `Centroid of the polygon`
8 $R \leftarrow \begin{bmatrix} \cos(t) & -\sin(t) \\ \sin(t) & \cos(t) \end{bmatrix}$ ▷ `Rotation matrix`
9 $rotatedPolygonVertices \leftarrow (polygonVertices - centroid) \cdot R^T + centroid$
10 $xCoords \leftarrow rotatedPolygonVertices(:, 1)$ ▷ `X coordinates`
11 $yCoords \leftarrow rotatedPolygonVertices(:, 2)$ ▷ `Y coordinates`
12 $minX \leftarrow \min(xCoords)$, $maxX \leftarrow \max(xCoords)$
$minY \leftarrow \min(yCoords)$, $maxY \leftarrow \max(yCoords)$
13 $flightPaths \leftarrow []$, $waypoints \leftarrow []$, $totalTurns \leftarrow 0$
14 **for** $y \in minY : Sidelap : maxY$ **do**
15 **if** $mod(i, 2) = 1$ **then**
16 `// Left to right`
17 $xLine \leftarrow minX : EndLap : maxX$
18 **end**
19 **else**
20 `// Right to left`
21 $xLine \leftarrow maxX : -EndLap : minX$
22 **end**
23 $linePoints \leftarrow [xLine', repmat(y, length(xLine), 1)]$
24 $inPolygon \leftarrow inpolygon(linePoints(:, 1), linePoints(:, 2), rotatedPolygonVertices(:, 1), rotatedPolygonVertices(:, 2))$
$filteredPoints \leftarrow linePoints(inPolygon, :)$
25 **if** $filteredPoints \neq []$ **then**
26 $flightPaths \leftarrow flightPaths \cup filteredPoints$
$waypoints \leftarrow waypoints \cup filteredPoints$
$totalTurns \leftarrow totalTurns + 1$
27 **end**
28 **end**
29 **if** $totalTurns > 0$ **then**
30 $totalTurns \leftarrow totalTurns - 1$
31 **end**
32 $waypoints \leftarrow (waypoints - centroid) \cdot R + centroid$ **return** $flightPaths, waypoints, totalTurns$

Figure 2. *The pseudo-code of the FMP algorithm, proposed for computing waypoints and total turns.*

Endlap and *Sidelap* given in lines #5–6 and lines #14–28 of Fig. 2 analytically and compactly describe the processes for constructing the interlacing path, calculating waypoint positions, and determining the total number of turns.

In this paper, the relevant FMP problem has been solved separately for four different scenarios.

https://doi.org/10.1017/aer.2025.10030 Published online by Cambridge University Press

The endurance of micro-UAVs in the air is generally limited due to the restricted capacity of their on-board energy sources. Since the flight endurance is a function of the total flight path length, the classical approach for FMP aims to minimise the total flight path length. The first FMP problem is based on minimising the total flight path length. This optimisation was achieved by applying Equation (6).

$$ObjFun: \underbrace{\arg\min}_{\text{azimuth}} \ \text{TotalDistance} = \sum_{i=1}^{n-1} \sqrt{(x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2 + (z_{i+1}-z_i)^2}, \tag{6}$$

where, the flight path calculated as a result of the FMP is represented by a list of waypoints, $(x, y, z)_i$ indicates the 3D spatial position of the i^{th} waypoint.

In aerial imaging-based *a-SfM* applications utilising micro-UAVs, the accuracy levels of the calculated topographic surface points may exhibit undesirable local fluctuations due to variations in GSD values. Therefore, it may be necessary to prepare the FMP in a manner that ensures relatively stable GSD values along the waypoints. The second FMP planning problem aims to find the narrowest 95% confidence interval for the GSD values calculated for the image capture points along the flight path. This ensures that the GSD values for the image capture points, and consequently the flight path, have the minimum average and standard deviation. This problem is modelled by Equation (7).

$$ObjFun: \underbrace{\arg\min}_{azimuth} \ (\mu_{GSD} + 3 \cdot \sigma_{GSD}) \tag{7}$$

where $[\mu, \sigma]_{GSD} = [mean\ (GSD)\ ,\ std\ (GSD)]$ | $GSD = \left[GSD_{(x,y,z)_i}\right]$ denote the mean and standard deviation of GSD values.

The turning manoeuvres of micro-UAVs involve complex processes, and the efficiency of on-board energy usage increases as the number of turns decreases. In *a-SfM* applications based on micro-UAV-supported aerial imaging, the accuracy levels of the calculated topographic surface points may exhibit undesirable numerical fluctuations due to the influence of GSD values by local topographic height variations. Therefore, it may be necessary to prepare an FMP that ensures a relatively stable GSD value along the waypoints while minimising the number of turns. The third FMP problem aims to minimise the 95% confidence interval for the GSD values calculated for the image capture points along the flight path and to minimise the number of required turns. This problem is modelled using Equation (8).

$$ObjFun: \underbrace{\arg\min}_{azimuth} \ [(\mu_{GSD} + 3 \cdot \sigma_{GSD}) + totalTurns]\,. \tag{8}$$

Micro-UAVs commonly utilise the *Stop-Turn-Go* strategy to complete their turning manoeuvres. The number of turning manoeuvres is a result of the FMP. Therefore, it may be necessary to prepare an FMP that minimises the number of turns for energy conservation. The fourth FMP planning problem aims to minimise the total number of turns, *totalTurns* (see lines #25-#31 of the Fig. 2). This problem is modelled using Equation (9).

$$ObjFun: \underbrace{\arg\min}_{azimuth} \ totalTurns \tag{9}$$

In this paper, CSA has been employed to address the planning problems formulated in Equations (6)–(9) for determining the optimal *azimuth* value. Fundamentally, CSA is responsible for computing the *azimuth* angle of the flight paths as defined in line #8 of Fig. 2. Once this *azimuth* angle is determined, the flight plan parameters that satisfy the corresponding geometric constraints can be calculated using the algorithm presented in Fig. 2.

3.0 Colony-based search algorithm (CSA)

The CSA [51] was developed as an evolutionary-based global minimiser tailored for addressing single-objective, bounded or unbounded real-valued numerical problems. Within the current iteration, CSA

https://doi.org/10.1017/aer.2025.10030 Published online by Cambridge University Press

focuses solely on evolving the patterns within the clan matrix, which is formed by randomly chosen patterns from a parent population referred to as the colony matrix. CSA operates as a non-recursive, iterative and reasonably robust approach. Its framework comprises initialisation, selection, numerical evolution and updating phases. The mutation mechanism in CSA is integrated with the random crossover process, collectively termed the morphogenesis process. The algorithmic design of CSA is thoroughly elaborated, step-by-step, in the subsequent descriptions.

The primary population of CSA, termed the Colony Matrix, $p0$, comprises a random solution vector with dimensions equivalent to the Clan Matrix, p, multiplied by T time. The initial Colony Matrix, $p0$, along with the objective function values of the Colony patterns, $fitp0$, are determined through the methodology outlined in Equation (10).

$$\begin{cases} p0_{i=1:T\cdot N, j=1:D} \sim U(low_j, up_j) \\ fitp0 = \text{ObjFun}\,(p0) \end{cases}, \tag{10}$$

where *ObjFun*, *low*, and *up* denote the objective function, the lower and upper bounds of the *search space*, respectively. In this study, the Mersenne Twister uniform pseudo-random number generator is employed to produce random numbers utilised across all experimental methods. In CSA, the initial values of *moment* and *initindex* are set to $moment_{initindex} = 0$ and $initindex = \{1{:}N\}$, respectively.

Equation (11) can be used to express the clan, p, and objective function values of the p, $fitp$, patterns.

$$[p;\ fitp] = [p0(index, :);\ fitp0(index)] \tag{11}$$

where $index = rprm\,(T \cdot N; N) \mid (initindex \sim = inex)$ and updated $initindex = index$. Equation (12) is used to generate the scale factor, or *scale*, that the CSA uses to regulate the direction vector's amplitude:

$$scale = \begin{cases} (rand\,(N; c) - 0.50) \oslash (rand\,(N, c) - 0.50) & If\ \ rand() < rand() \\ sign\,(rand\,(N; 1) - 0.50) \circ \Phi & else \end{cases} \tag{12}$$

where $\Phi \sim LevyDist\,(\alpha, \beta, N, 1)$ and $\oslash$ denote scaling vector, and Hadamard division operator, respectively. The individuals of Φ have Levy distribution with α, β location and shape parameters. Φ supplies Levy flight ability to CSA. In the CSA framework, Levy flights provide the essential exploration capability needed to overcome local optima and identify a variety of high-quality solutions within intricate search spaces. By incorporating sporadic long jumps, Levy flights strengthen the global search process, enhancing algorithm robustness and reducing the likelihood of stagnation. The Levy distribution, distinguished by its heavy-tailed nature, is a probability distribution that exhibits a greater tendency to generate extreme values than the normal distribution. It is characterised by its probability density function (PDF), as specified in Equation (13).

$$f(x; \alpha, \beta) = \frac{\beta}{2\pi} \exp\left(-\frac{\beta}{2\,(x-\alpha)}\right)\left(-\frac{1}{(x-\alpha)^{3/2}}\right) \tag{13}$$

where, x, α and β denote the random variable, the location parameter, and the scale parameter of the distribution, respectively. In CSA, the Φ used in Equation (12) is generated using Equation (14):

$$\Phi = \beta \cdot (rand() + 1) \oslash \omega^{\frac{1}{\alpha}} \quad \mid \quad \omega \sim \Gamma\,(\alpha; \kappa)\,,\ \ \kappa \sim \{2:5\} \tag{14}$$

where $\Gamma\,(\alpha; \kappa)$ denotes Gamma distribution with the parameters of $(\alpha; \kappa)$.

The value of c used in Equation (12) is computed using Equation (15):

$$If\ \ rand() < rand()\ ,\ \ then\ \ c = 1\ \ \ else\ \ \ c = D \tag{15}$$

https://doi.org/10.1017/aer.2025.10030 Published online by Cambridge University Press

Using the steps outlined in Equation (16), the binary valued *mutation control matrix*, m, of CSA is produced.

$$\begin{cases} m = 0_{N\times D} \\ for\ \ j = 1\ \ to\ \ N \\ ind = rperm\ (D) \\ k_{index} = abs(randi([0,\ 1]) - rand()^{\bullet randi([1,10])}) \\ e = ind\ (1 : \lceil k_{index} \cdot D \rceil) \\ m\ (k, e) := 1 \\ end \end{cases} \tag{16}$$

where $\bullet$ denotes piecewise-power operator. CSA uses Equation (17) to generate evolutionary direction, *dx*.

$$dx = \begin{cases} p\left[v2\right] - p\left[v1\right] & v = 1 \\ p[v1] - p & v = 2 \\ p\left[index0\left[randi\ (1, N/5)\right], :\right] - p & v = 3 \end{cases}, \tag{17}$$

where $[v1, v2] = \left[rperm\ (N; N)\ , rperm\ (N; N)\right]$, $v1 \neq v2$ and $index0 = sortindex\ (fit, 'ascend') \ \ \wedge \ \ v \sim U\ \{1, 2, 3\}$.

The morphogenesis matrix, *px*, of CSA is generated using Equation (18):

$$px = p + scale \circ m \circ dx + s \tag{18}$$

where $s = (rand\ (N; 1) - 0.50) \circ rand(N; 1)^{\bullet randi([2;10])}$.

In CSA, Equation (19) is used to update the individuals that exceed the search boundaries:

$$\begin{aligned} &for\ \ k = 1\ \ to\ \ N \\ &for\ \ l = 1\ \ to\ \ D \\ &If\ \ px(k, l) < low(l),\ \ px(k, l) = low(l) + rand()^{randi([1;5])} \cdot (up(l) - low(l)) \\ &If\ \ px(k, l) > up(l),\ \ px(k, l) = up(l) + rand()^{randi([1;5])} \cdot (low(l) - up(l)) \\ &end \\ &end \end{aligned} \tag{19}$$

The objective function values of *px* patterns are computed using Equation (20):

$$fitpx = \text{ObjFun}(px) \tag{20}$$

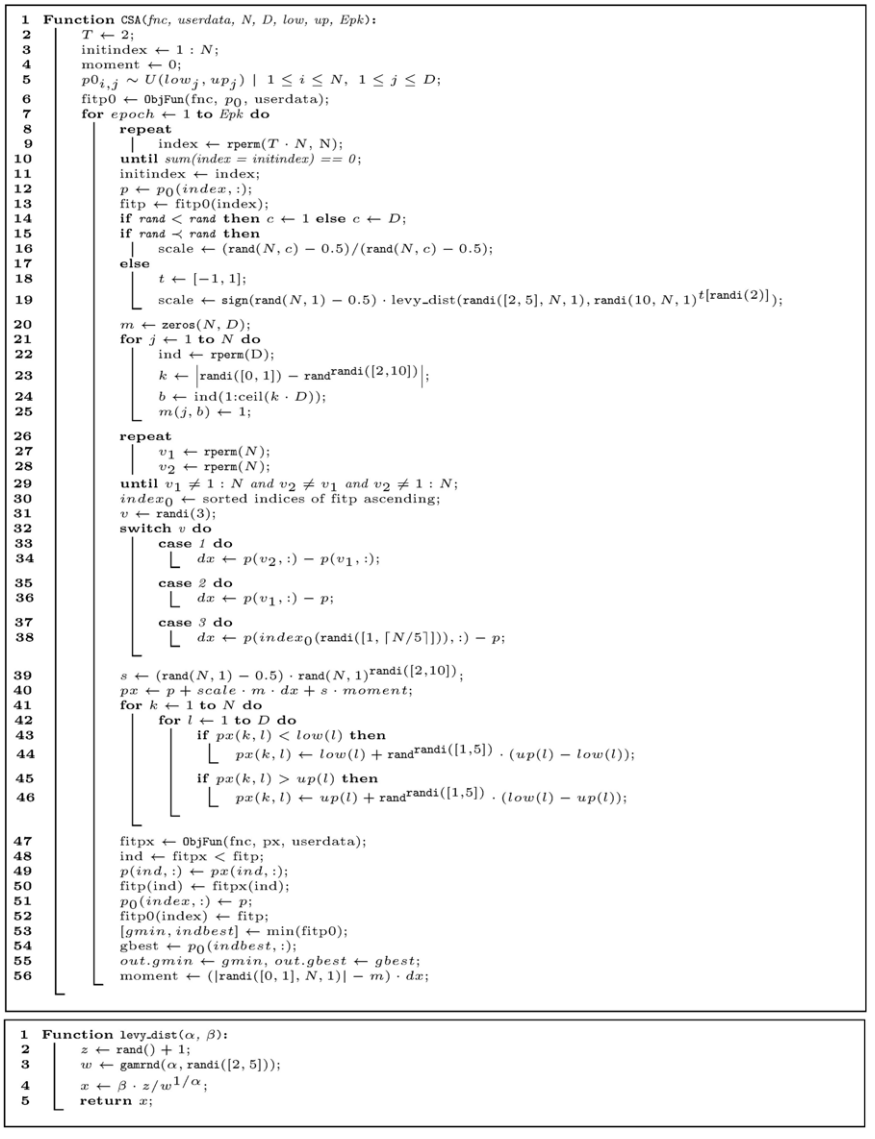
The orders, *ind* of the *px* patterns supply better solutions then the corresponding patterns are obtained using Equation (21):

$$ind \leftarrow fitpx_{\{1:N\}} < fitp_{\{1:N\}} \ \ | \ \ \ ind \in \{1 : N\} \tag{21}$$

In the concluding stage of each iteration, CSA modifies the associated colony patterns by incorporating the clan patterns as outlined in Equation (22).

$$\begin{cases} \left[p(ind, :)\ , fitp(ind)\right] = \left[px(ind, :)\ , fitpx(ind)\right] \\ [p0(index, :), fit0(index)] = \left[p, fitp\right] \\ indbest \leftarrow fitp0_{indbest} = \min(fitp0) \\ [gbest, gmin] = \left[p0(indbest, :)\ , fit0(indbest)\right] \end{cases}. \tag{22}$$

https://doi.org/10.1017/aer.2025.10030 Published online by Cambridge University Press



```
Function CSA(fnc, userdata, N, D, low, up, Epk):
    T ← 2;
    initindex ← 1 : N;
    moment ← 0;
    p0_{i,j} ~ U(low_j, up_j) | 1 ≤ i ≤ N, 1 ≤ j ≤ D;
    fitp0 ← ObjFun(fnc, p_0, userdata);
    for epoch ← 1 to Epk do
        repeat
            index ← rperm(T · N, N);
        until sum(index = initindex) == 0;
        initindex ← index;
        p ← p_0(index, :);
        fitp ← fitp0(index);
        if rand < rand then c ← 1 else c ← D;
        if rand ≺ rand then
            scale ← (rand(N, c) − 0.5)/(rand(N, c) − 0.5);
        else
            t ← [−1, 1];
            scale ← sign(rand(N, 1) − 0.5) · levy_dist(randi([2, 5], N, 1), randi(10, N, 1)^{t[randi(2)]});
        m ← zeros(N, D);
        for j ← 1 to N do
            ind ← rperm(D);
            k ← |randi([0, 1]) − rand^{randi([2,10])}|;
            b ← ind(1:ceil(k · D));
            m(j, b) ← 1;
        repeat
            v_1 ← rperm(N);
            v_2 ← rperm(N);
        until v_1 ≠ 1 : N and v_2 ≠ v_1 and v_2 ≠ 1 : N;
        index_0 ← sorted indices of fitp ascending;
        v ← randi(3);
        switch v do
            case 1 do
                dx ← p(v_2, :) − p(v_1, :);
            case 2 do
                dx ← p(v_1, :) − p;
            case 3 do
                dx ← p(index_0(randi([1, ⌈N/5⌉])), :) − p;
        s ← (rand(N, 1) − 0.5) · rand(N, 1)^{randi([2,10])};
        px ← p + scale · m · dx + s · moment;
        for k ← 1 to N do
            for l ← 1 to D do
                if px(k, l) < low(l) then
                    px(k, l) ← low(l) + rand^{randi([1,5])} · (up(l) − low(l));
                if px(k, l) > up(l) then
                    px(k, l) ← up(l) + rand^{randi([1,5])} · (low(l) − up(l));
        fitpx ← ObjFun(fnc, px, userdata);
        ind ← fitpx < fitp;
        p(ind, :) ← px(ind, :);
        fitp(ind) ← fitpx(ind);
        p_0(index, :) ← p;
        fitp0(index) ← fitp;
        [gmin, indbest] ← min(fitp0);
        gbest ← p_0(indbest, :);
        out.gmin ← gmin, out.gbest ← gbest;
        moment ← (|randi([0, 1], N, 1)| − m) · dx;

Function levy_dist(α, β):
    z ← rand() + 1;
    w ← gamrnd(α, randi([2, 5]));
    x ← β · z/w^{1/α};
    return x;
```

Figure 3. *Pseudo-code of CSA with the sub-function levy_dist.*

At the conclusion of the present iteration, the numerical value of is revised according to Equation (23):

$$moment = (abs(randi([0,1]; N, 1)) - m) \circ dx \tag{23}$$

The pseudo algorithm of the CSA has been illustrated in Fig. 3.

CSA is an iterative, non-recursive, stochastic evolutionary search method. CSA is developed for solving single-objective, real valued numerical problems. The version of CSA presented in this paper is designed as a global minimiser. CSA scales the amplitude of the direction vectors using a unique scaling method.

https://doi.org/10.1017/aer.2025.10030 Published online by Cambridge University Press

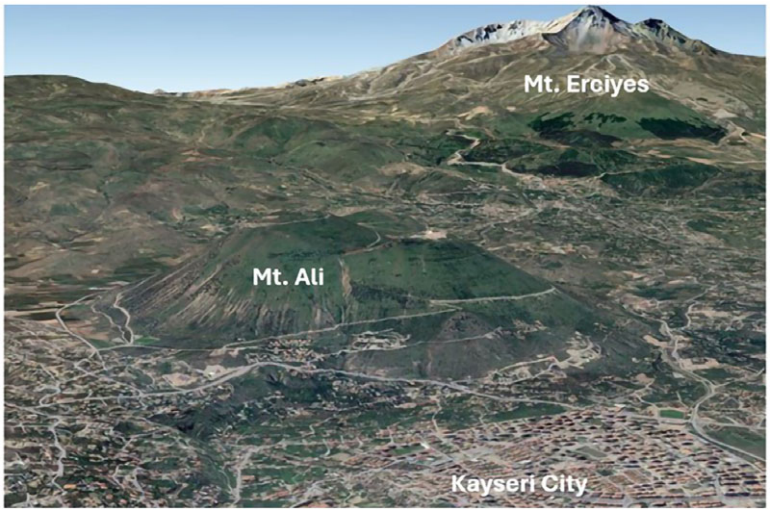


Figure 4. *Mt. Ali and Mt. Erciyes.*

4.0 Experiments

In the experiments conducted in this section, four distinct optimal FMP were calculated using the following parameter values: $p = 0.80$, $q = 0.70$, $H = 200$ m, $ImageWidth = 5472$ pixels, $ImageHeight = 3648$ pixels, $F = 8.8$ mm, and $ccdWidth = 2.41\ \mu$ m. The flight plans were designed over a test polygon that partially covers Mt. Ali, with geographical coordinates (lat=38°.6621, Lon=35°.5533, WGS84). Figure 4 visualises the general view of test area, Mt. Ali.

The EPSG:3395 (WGS84/World Mercator) projection system was used for visualising the flight plans obtained from the conducted experiments. The EPSG:3395, also known as the WGS84/World Mercator projection, is a widely used mapping system that provides a conformal representation of the Earth's surface. This projection system preserves angles and shapes over small areas, making it particularly suitable for applications requiring accurate directional relationships, such as navigation and flight path visualisation. One of its key advantages is its ability to represent large-scale maps with minimal distortion near the equator, although distortions increase at higher latitudes. The uniformity of scale along the equator and its compatibility with global datasets make EPSG:3395 a preferred choice for tasks involving geospatial analysis and cartographic rendering. Additionally, the projection's grid-based structure simplifies calculations for mapping purposes, enhancing computational efficiency. Its widespread adoption in geographic information systems (GIS) ensures seamless integration with existing tools and frameworks. Another notable benefit is its consistent representation of longitudinal lines as vertical and latitudinal lines as horizontal, which aids in the interpretation of spatial data. These characteristics contribute to its reliability and usability in applications where precise spatial alignment is critical.

The local altitude, H, of the micro-UAV for the $(i, j)^{\text{th}}$ pixel location of the digital terrain model (DTM) has been fixed as $H = 200\,m$. The highest geodesic peak within the flight polygon is $h_0 = \max(DTM) = 1851.44m$. Thus, the micro-UAV flies at a steady elevation of $h_{\max} = h_0 + 200 = 2051.44\,m$ throughout the flight.

In this paper, the digital elevation model (DEM) data of Mt. Ali, obtained from the ASTER GDEM v3 global elevation dataset [60] with a spatial resolution of 1 arc-second, was used to generate the GSD map presented in Fig. 5.

In this paper, each of the four distinct FMP problems addressed contains a single unknown corresponding to the *azimuth* value. To determine the optimal value of (*azimuth*), the CSA was employed. The control parameters of CSA were configured as follows: clan size $N = 30$, problem dimension $D = 1$, maximum number of iterations $MaxCycle = 100,000$, lower bounds low $= [0, -100, -100]$ and upper

https://doi.org/10.1017/aer.2025.10030 Published online by Cambridge University Press

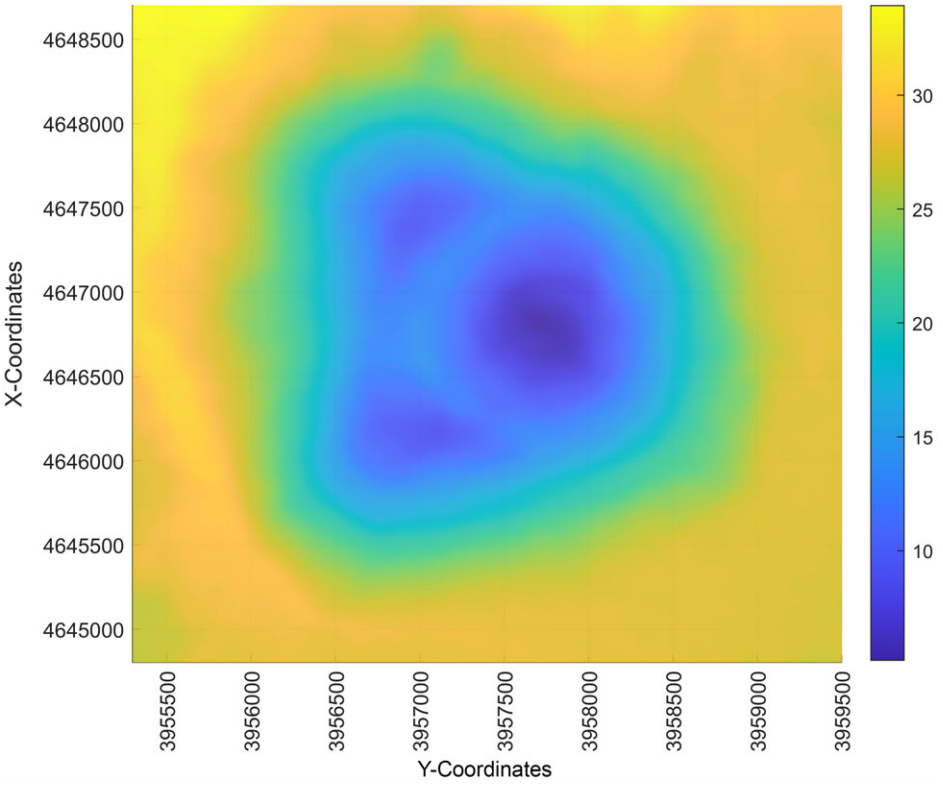


Figure 5. *GSD map for* $h_{\max} = 2051.44$ m. *and* $H = 200$ *m.*

bounds up = [360, 200, 200]. These values were selected empirically. CSA is not overly sensitive to the initial parameter settings.

The first FMP problem (Experiment #1), which aims to minimise the total flight distance, is defined using Equation (6). In this experiment, a global solution value of $azimuth = 91^0.06024$ was obtained. According to the obtained global solution, the total flight distance is 33,432.55 m. In this scenario, a total of 29 turns were made. A total of 789 image capture point locations were calculated. The average GSD is 14.80, and the standard deviation value is 3.28.

The second FMP problem (Experiment #2), as given in Equation (7), focuses on achieving the narrowest 95% confidence interval for the GSD values at the image capture points along the flight path. In this experiment, a global solution value of $azimuth = 268^0.6806$ was obtained. According to the obtained global solution, the total flight distance is 34051.11 m. In this scenario, a total of 30 turns were made. A total of 800 image capture point locations were calculated. The average GSD is 14.79, and the standard deviation value is 3.26.

The third FMP problem (Experiment #3), as given in Equation (8), aims to minimise both the 95% confidence interval for the GSD values calculated at the image capture points along the flight path and the number of required turns. In this experiment, a global solution value of $azimuth = 199^0.1012$ was obtained. According to the obtained global solution, the total flight distance is 35574.54 m. In this scenario, a total of 30 turns were made. A total of 819 image capture point locations were calculated. The average GSD is 14.93, and the standard deviation value is 3.37.

The fourth FMP problem (Experiment #4), as defined in Equation (9), aims to minimise the total number of turns. In this experiment, a global solution value of $azimuth = 47^0.8804$ was obtained. According to the obtained global solution, the total flight distance is 36232.08 m. In this scenario, a total of 35 turns

https://doi.org/10.1017/aer.2025.10030 Published online by Cambridge University Press

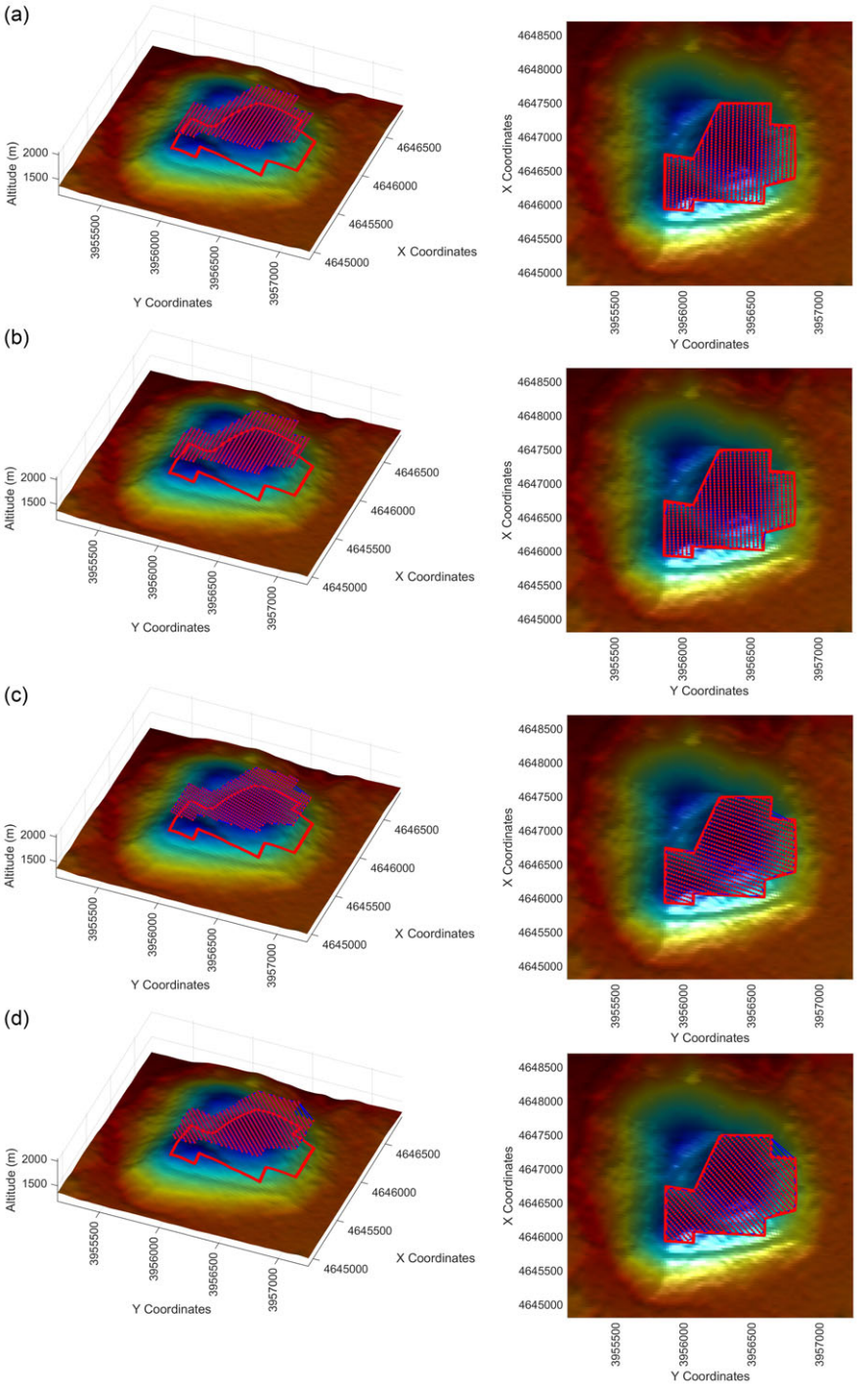


Figure 6. *Visualisation of experimental results of : (a) Experiment #1, (b) Experiment #2, (c) Experiment #3 and (d) Experiment #4.*

https://doi.org/10.1017/aer.2025.10030 Published online by Cambridge University Press

were made. A total of 819 image capture point locations were calculated. The average GSD is 14.95, and the standard deviation value is 3.40. The results obtained from Experiments #1–4 are illustrated in Fig. 6.

5.0 Conclusion

Small-scale FMP applications frequently utilise micro-UAV's, which are constrained by limited on-board energy capacities. This limitation makes energy efficiency a critical factor in mission planning. Among the most influential parameters affecting energy consumption are the total length of the flight path and the number of images captured during the mission. Micro drones generally lack sophisticated speed control systems, meaning that reducing flight time is largely dependent on minimising the overall flight path length. Each image capture process consumes a portion of the drone's on-board energy resources, further underscoring the importance of optimising both the flight plan and the imaging process. Designing FMP under geometric constraints, such as image overlap ratios and GSD, plays a vital role in ensuring operational safety, efficiency and high-quality outputs. This study introduces four problem models corresponding to different scenarios for calculating optimal flight paths under varying constraints, offering a systematic approach to flight planning. The optimisation problems were solved using the CSA algorithm, demonstrating its capability to address complex flight path design challenges effectively. The results reveal that analytical flight planning methods outperform manual planning based on user experience, particularly in terms of flight safety, energy efficiency and image quality. Analytical approaches provide objective and reliable solutions, reducing the risk of human error associated with subjective judgement. Optimised flight paths significantly decrease unnecessary energy consumption, thereby extending the drone's operational lifespan and improving mission feasibility. By minimising the total flight distance, the proposed models not only reduce mission duration but also lower operational costs, making them highly practical for small-scale *a-SfM*, such as aerial mapping. Efficient flight planning ensures that the drone can complete its mission without the risk of sudden power depletion, which is particularly critical for micro drones with limited battery capacity. Geometric constraints such as GSD and overlap ratios are essential for achieving high-quality *a-SfM* outputs, and their integration into the optimisation process guarantees that the generated flight plans meet both operational and quality standards. Manual flight planning often leads to suboptimal paths due to reliance on subjective judgement, whereas analytical methods provide objective and data-driven solutions.

The proposed optimisation framework highlights the potential for enhancing the reliability of FMP conducted with micro drones, particularly in resource-constrained environments. Reducing the number of turns in the flight path not only conserves energy but also minimises the likelihood of navigational errors, contributing to smoother and more efficient operations. This study emphasises the importance of adopting systematic and analytical approaches to improve the performance of small-scale aerial mapping, particularly when using energy-limited platforms. The findings suggest that optimising flight paths can lead to significant improvements in mission efficiency, especially in scenarios where energy resources are scarce.

Future research could explore additional constraints and variables to further refine the optimisation models, expanding their applicability to a broader range of aerial imaging scenarios. For instance, incorporating environmental factors such as wind speed or terrain elevation could enhance the robustness of the flight planning process. Investigating the integration of real-time data during missions might also enable dynamic adjustments to flight paths, further improving efficiency and adaptability. Additionally, the development of user-friendly tools for implementing these optimisation models could facilitate their adoption by practitioners in the field. Overall, this study demonstrates the value of analytical flight planning in achieving safer, more efficient and higher-quality FMP, paving the way for advancements in small-scale drone-based mapping and surveying.

https://doi.org/10.1017/aer.2025.10030 Published online by Cambridge University Press

References

[1] Dancila, B.D. and Botez, R.M. Vertical flight path segments sets for aircraft flight plan prediction and optimisation, *Aeronaut J*, 2018, **122**, (1255), pp 1371–1424.

[2] Guo, A., Anderson, D. and Botez, R.M. Vector field path following of a full-wing solar-powered Unmanned Aerial Vehicle (UAV) landing based on Dubins path: a lesson from multiple landing failures, *Aeronaut J*, 2021, **125**, (1290), pp 1283–1312.

[3] Gutierrez-Martinez, M.A., et al. Genetic algorithm-based path planning of quadrotor UAVs on a 3D environment, *Aeronaut J*, **129**, (1334), 2025, pp 902–938.

[4] Hall, J. and Anderson, D. Reactive route selection from pre-calculated trajectories – application to micro-UAV path planning, *Aeronaut J*, 2011, **115**, (1172), pp 635–640.

[5] Hameed, R.U., et al. Reinforcement learning-based radar-evasive path planning: a comparative analysis, *Aeronaut J*, 2022, **126**, (1297), pp 547–564.

[6] Thoma, A., et al. Prioritising paths: An improved cost function for local path planning for UAV in medical applications, *Aeronaut J*, 2023, **127**, (1318), pp 2125–2142.

[7] Hao, G.Q., et al. UAV path planning based on improved artificial potential field method, *Aerospace*, 2023, **10**, (6), p 562.

[8] Huang, J.H., et al. A practical interlacing-based coverage path planning method for fixed-wing UAV photogrammetry in convex polygon regions, *Aerospace*, 2022, **9**, (9), p 521.

[9] Li, J.C., et al. Coverage path planning method for agricultural spraying UAV in arbitrary polygon area, *Aerospace*, 2023, **10**, (9), p 755.

[10] Chai, X.Z., et al. Multi-strategy fusion differential evolution algorithm for UAV path planning in complex environment, *Aerosp Sci Technol*, 2022, **121**, p 107287.

[11] Huang, Y., Wang, H.L. and Yao, P. Energy-optimal path planning for Solar-powered UAV with tracking moving ground target, *Aerosp Sci Technol*, 2016, **53**, pp 241–251.

[12] Pehlivanoglu, Y.V., Baysal, O. and Hacioglu, A. Path planning for autonomous UAV via vibrational genetic algorithm, *Aircr Eng Aerosp Technol*, 2007, **79**, (4), pp 352–359.

[13] Zhou, G.L., et al. Path Planning of UAV Using Levy Pelican Optimization Algorithm In Mountain Environment, *Appl Artif Intell*, 2024, **38**, (1), pp 2368343.

[14] Simoes, D.P., Oliveira, H.C. and Garcia, M.V.Y. UAV 3-D Path Planning Based on High-Resolution DSM, DTM, and True Orthomosaic, *IEEE Geosci Remote Sens Lett*, 2022, **19**, pp 1–5.

[15] Salach, A., et al. Accuracy assessment of point clouds from LiDAR and dense image matching acquired using the UAV platform for DTM creation, *ISPRS Int J Geo-Inf*, 2018, **7**, (9), p 342.

[16] Maguya, A.S., Junttila, V. and Kauranne, T. Adaptive algorithm for large scale DTM interpolation from lidar data for forestry applications in steep forested terrain, *ISPRS J Photogramm Remote Sens*, 2013, **85**, pp 74–83.

[17] Cabo, C., et al. Influence of the number and spatial distribution of ground control points in the accuracy of UAV-SfM DEMs: An approach based on generalized additive models, *IEEE Trans Geosci Remote Sens*, 2021, **59**, (12), pp 10618–10627.

[18] Zhang, K., et al. An enhanced multi-view vertical line locus matching algorithm of object space ground primitives based on positioning consistency for aerial and space images, *ISPRS J Photogramm Remote Sens*, 2018, **139**, pp 241–254.

[19] Wang, Y.X., et al. 6-DOF UAV Path planning and tracking control for obstacle avoidance: a deep learning-based integrated approach, *Aerosp Sci Technol*, 2024, **151**, pp 109320.

[20] Puente-Castro, A., et al. UAV swarm path planning with reinforcement learning for field prospecting, *Appl Intell*, 2022, **52**, (12), pp 14101–14118.

[21] Feng, R.H., et al. Resource allocation and path planning for RSMA-enabled UAV-VLC networks considering UAV jitter, *Appl Opt*, 2025, **64**, (8), pp 1809–1820.

[22] Cong, W.Y., et al. An improved pied kingfisher optimizer for maritime UAV path planning, *Appl Sci-Basel*, 2024, **14**, (24), p 11816.

[23] Li, J., et al. RJA-star algorithm for UAV path planning based on improved R5DOS model, *Appl Sci-Basel*, 2023, **13**, (2), p 1105.

[24] Ortiz-Sanz, J., Bastos, G. and Gil-Docampo, M. Deformation measurement of twisted timber beam using UAV SfM photogrammetry and a new feature extraction algorithm, *Eur J Wood Wood Prod*, 2025, **83**, (2), p 88.

[25] Martínez-Fernández, A., et al. The influence of image properties on high-detail SfM photogrammetric surveys of complex geometric landforms: The application of a consumer-grade UAV camera in a rock glacier survey, *Remote Sens*, 2022, **14**, (15), p 3528.

[26] You, G.P., et al. Mixed-strategy Harris Hawk optimization algorithm for UAV path planning and engineering applications, *Appl Sci-Basel*, 2024, **14**, (22), p 10581.

[27] Yu, Y. and Lee, S.H. Efficient multi-UAV path planning for collaborative area search operations, *Appl Sci-Basel*, 2023, **13**, (15), p 8728.

[28] Zhang, N., et al. A UAV coverage path planning method based on a diameter-height model for mountainous terrain, *Appl Sci-Basel*, 2025, **15**, (4), p 1988.

[29] Guo, C., Huang, L. and Tian, K. Combinatorial optimization for UAV swarm path planning and task assignment in multi-obstacle battlefield environment, *Appl Soft Comput*, 2025, **171**, p 112773.

[30] Civicioglu, P. and Besdok, E. A plus Evolutionary search algorithm and QR decomposition based rotation invariant crossover operator. Exp Syst Appl, 2018, 103, pp 49–62.

[31] Ouyang, J., et al. Optimization of beamforming and path planning for UAV-assisted wireless relay networks, *Chin J Aeronaut*, 2014, **27**, (2), pp 313–320.

https://doi.org/10.1017/aer.2025.10030 Published online by Cambridge University Press

[32] Gong, C., et al. Energy-efficient task migration and path planning in UAV-enabled mobile edge computing system, *Complexity*, 2022, **2022**, p 269102.
[33] Zhang, N., et al. Large-area coverage path planning method based on vehicle-UAV collaboration, *Applied Sciences-Basel*, 2025, **15**, (3), p 1247.
[34] Yang, J.Q., et al. Hybrid chaos game and grey wolf optimization algorithms for UAV path planning, *Appl Math Modell*, 2025, **142**, p 115979.
[35] Pan, S.H., et al. Optimal coverage path planning for UAV-assisted multiple USVs: Map modeling and solutions, *Drones*, 2025, **9**, (1), p 30.
[36] Konar, M., Cam, O. and Aktas, M.O. BSO algorithm and artificial neural network aided emission optimisation for gas turbine engine, *Aeronaut J*, 2024, **129**, pp 107–122.
[37] Konar, M., Hatipoglu, S.A. and Akpinar, M. Improvement of UAV thrust using the BSO algorithm-based ANFIS model, *Aeronaut J*, 2024, **128**, (1328), pp 2364–2373.
[38] Gunen, M.A., et al. Camera calibration by using weighted differential evolution algorithm: a comparative study with ABC, PSO, COBIDE, DE, CS, GWO, TLBO, MVMO, FOA, LSHADE, ZHANG and BOUGUET, *Neural Comput Appl*, 2020, **32**, (23), pp 17681–17701.
[39] Dhruva, A., et al. Effective UAV photogrammetry for forest management: New insights on side overlap and flight parameters, *Forests*, 2024, **15**, (12), p 2135.
[40] Domingo, D., et al. Effects of UAV image resolution, camera type, and image overlap on accuracy of biomass predictions in a tropical woodland, *Remote Sens*, 2019, **11**, (8), p 948.
[41] Liu, P., et al. Improved adaptive snake optimization algorithm with application to multi-UAV path planning, *Trans Inst Meas Control*, 2024, **47**, (8), pp 1639–1650. https://doi.org/10.1177/01423312241263637
[42] Liu, J., et al. Agricultural UAV path planning based on a differentiated creative search algorithm with multi-strategy improvement, *Machines*, 2024, **12**, (9), p 591.
[43] Liu, B., et al. A hybrid ARO algorithm and key point retention strategy trajectory optimization for UAV path planning, *Drones*, 2024, **8**, (11), p 644.
[44] Liang, H.J., et al. Solving UAV 3D path planning based on the improved lemur optimizer algorithm, *Biomimetics*, 2024, **9**, (11), p 654.
[45] Zhu, Z.F., Hu, J.L., Wen, J.J. Evaluation of quantitative accuracy of obstacle status for substation UAV inspection, *Comput. Simul.*, 2022, **39**, p 387–391.
[46] Fu, S.W., et al. Red-billed blue magpie optimizer: A novel metaheuristic algorithm for 2D/3D UAV path planning and engineering design problems, *Artif Intell Rev*, 2024, **57**, (6), p 134.
[47] Zheng, J.B., et al. Distributed stochastic algorithm based on enhanced genetic algorithm for path planning of multi-UAV cooperative area search, *IEEE Trans Intell Transp Syst*, 2023, **24**, (8), pp 8290–8303.
[48] Ma, W., et al. Cooperative localisation of UAV swarm based on adaptive SA-PSO algorithm, *Aeronaut J*, 2023, **127**, (1307), pp 57–75.
[49] Liu, H.Y., et al. UAV swarm collaborative coverage control using GV division and planning algorithm, *Aeronaut J*, 2023, **127**, (1309), pp 446–465.
[50] Yao, J.Y., et al. IHSSAO: An improved hybrid Salp swarm algorithm and aquila optimizer for UAV path planning in complex terrain, *Appl Sci-Basel*, 2022, **12**, (11), p 5634.
[51] Civicioglu, P. and Besdok, E. Colony-based search algorithm for numerical optimization, *Appl Soft Comput*, 2024, **151**, p 111162.
[52] Civicioglu, P. and Besdok, E. Bernstein-Levy differential evolution algorithm for numerical function optimization, *Neural Comput Appl*, 2023, **35**, (9), pp 6603–6621.
[53] Civicioglu, P. Backtracking Search Optimization Algorithm for numerical optimization problems, *Appl Math Comput*, 2013, **219**, (15), pp 8121–8144.
[54] Civicioglu, P. Artificial cooperative search algorithm for numerical optimization problems, *Inform Sci*, 2013, **229**, pp 58–76.
[55] Yang, Y.N., et al. GSDDet: Ground Sample Distance-Guided Object Detection for Remote Sensing Images, *IEEE Trans Geosci Remote Sens*, 2024, **62**, p 5626012.
[56] Lee, Y., Son, J., Kim, T. Determination of optimal ground sampling distance for matching GCP chips and satellite images, In *24th ISPRS Congress On Imaging Today, Foreseeing Tomorrow*, Nice, France, 2022.
[57] Peng, A.S., et al. Design of a ground sampling distance graphical user interface for an unmanned aerial vehicle system, In *53rd Annual Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, 2019.
[58] Lee, J.H. and Sull, S. Regression tree CNN for estimation of ground sampling distance based on floating-point representation, *Remote Sens*, 2019, **11**, (19), pp 2276.
[59] Turef, 2022, https://www.tusaga-aktif.gov.tr/
[60] https://asterweb.jpl.nasa.gov/gdem.asp

Cite this article: Civicioglu P. (2025). Optimal flight path planning for micro-UAV-based aerial imaging using the colony-based search algorithm. *The Aeronautical Journal*, **129**, 2827–2842. https://doi.org/10.1017/aer.2025.10030

https://doi.org/10.1017/aer.2025.10030 Published online by Cambridge University Press