The Aeronautical Journal (2026), **130**, pp. 741–761
doi:10.1017/aer.2025.10101

ROYAL AERONAUTICAL SOCIETY

RESEARCH ARTICLE

Multi-UAV cooperative path planning based on chaotic grey wolf optimiser

S. Zhang and L. Wang

School of Artificial Intelligence, Hebei University of Technology, Tianjin, China
Corresponding author: L. Wang; Email: wlk0228@163.com

Received: 14 April 2025; **Revised:** 25 October 2025; **Accepted:** 28 October 2025

Keywords: chaotic; GWO; multi-UAV cooperative; path planning

Abstract

The success of cooperative unmanned aerial vehicle (UAV) missions relies on effective multi-UAV path planning. To address the issues of limited individual diversity and susceptibility to local optima during the population initialisation phase of the traditional grey wolf optimiser (GWO), this paper proposes an improved chaotic grey wolf optimiser (CGWO). The algorithm enhances population diversity by introducing chaotic initialisation to generate more uniformly distributed initial solutions. Combined with a chaotic local search strategy, it employs a dynamic balancing mechanism that allows individuals in the population to strike a balance between global exploration and local refinement, thereby breaking free from local optima constraints and accelerating optimal solution convergence. The algorithm was validated by using the CEC2017 benchmark functions and simulations of multi-UAV mission scenarios. The results clearly demonstrate that the improved algorithm significantly outperforms the original GWO and other common optimisation algorithms in terms of convergence accuracy and speed during benchmark testing. In multi-UAV mission scenarios, the enhanced algorithm excels in trajectory planning, flexibly avoiding obstacles while maintaining smooth flight paths for UAVs. Overall, this research provides a practical solution for coordinated multi-UAV operational path planning.

Nomenclature

altitude cost

UAV	unmanned aerial vehicle
GWO	grey wolf optimiser
$CGWO$	chaotic grey wolf optimiser
S	origin
E	destination
d_T	horizontal distance from the UAV to the obstacle's centre
d_{Tmin}	minimum distance limit between the UAV and the obstacle
h_i	height of UAVs
h_{safe_i}	safety height
L_i	trajectory distance of the ith drone
v_{max}, v_{min}	maximum and minimum speed
K	number of peaks
m	number of sampling points
M	number of UAVs
n	number of waypoints
t	number of iterations
T	Maximum iterations
O_{fuel}	fuel cost
O_{threat}	threat cost

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

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

O_{angle}	angle cost
O_{height}	redaltitude cost
O_{time}	time cost
O_{dis}	distance cost

Greek symbol

α	the ratio of the overlapping time interval of the two UAVs
ω	weighting factor
θ	slope of the peak
β_i, φ	climbing angle and steering angle
μ	control parameter

1.0 Introduction

In the past few years, the swift advancements in UAV technology have revolutionised multiple sectors [1, 2]. From military surveillance and environmental assessments to traffic oversight and disaster relief efforts, UAVs have become indispensable tools across diverse industries [3, 4]. However, the efficient planning of UAV flight paths, especially in multi-UAV cooperative operations, remains a challenging research problem. Multi-UAV path planning not only requires individual UAVs to be able to accomplish their respective tasks, but must also consider variables like UAV teamwork, collision prevention and power usage. Therefore, how to realise efficient collaboration and trajectory optimisation for multiple UAVs in intricate settings has emerged as a key focus in contemporary UAV studies [5–8].

Nowadays commonly used multi-UAV path planning algorithms include traditional path planning algorithms based on graph search, intelligent optimisation algorithms and fusion algorithms based on reinforcement learning. Traditional path planning algorithms are mainly based on graph search, which mainly include Dijkstra algorithm, A-star algorithm [9], breadth-first search (BFS) algorithm [10] and rapidly exploring random trees (RRT) algorithm [11]. However, traditional road path planning algorithms are usually only applicable to simple scenarios and have poor computational efficiency and flexibility when facing dynamic obstacles and complex environments. Intelligent optimisation algorithms are modeled after natural or social phenomena that simulate their behaviour or mechanisms to solve optimisation problems. Compared with traditional algorithms, intelligent optimisation algorithms possess robust global search abilities, making them appropriate for intricate, nonlinear and large-scale optimisation challenges. Common intelligent optimisation algorithms include particle swarm optimisation algorithm (PSO) [12], genetic algorithm (GA) [13], differential evolutionary (DE) algorithm [14], immune plasma algorithm (IPA) [15], grey wolf optimisation (GWO) [16], and others. Currently, various meta-heuristic algorithms have been applied to solve drones' path planning problems. Reference [17] proposes a preference-based evolutionary algorithm for constrained multi-UAV path planning. Reference [18] combines PSO with an improved symbiotic method, thereby solving the problem of cooperative paths for multi-UAV. Reference [19] uses a newly proposed method called extended treatment approach to improve the IPA and applies it to aircraft path planning, resulting in a significant improvement in path quality.

The GWO is suitable for complex, nonlinear and unstructured optimisation problems, which not only possesses strong global search capability, but also low parameter adjustment demand, simple implementation and strong adaptability, which makes it has great flexibility in practical application. However, the GWO suffers from a problem that increases the probability of being confined to local optima and results in slower convergence. Reference [20] proposes a spiral position update technique using GWO, assigning the golden ratio as its probability parameter. Reference [21] introduces a nonlinear convergence factor and a levy on-the-fly update strategy to enhance the optimisation of the algorithm. Reference [22] combines GWO with DE, optimising the equilibrium of algorithmic evolution and search. Nonlinear transformations of parameters enhance algorithmic performance.

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

In this paper, CGWO is introduced to address multi-UAV path planning in intricate environments. In the initialisation phase, chaotic initialisation is used instead of random initialisation to enhance population diversity and stop the algorithm from converging to local optima. During the iteration of the algorithm, chaotic local search is introduced to improve local search diversity and exploration, preventing premature convergence to the local optimal solution. In addition, the leader dynamic reduction mechanism is used in the following phase to provide a balanced adaptive search for both global and local optimisation, enhancing the algorithm's convergence rate and precision.

2.0 Modeling cooperative path planning for multi-uav

During UAV route optimisation, we need to consider a variety of constraints. In multi-UAV path planning, the first step is to model the flight environment and establish an environment model based on it. Then, define S as the origin, E as the destination, with N intermediate points composing the UAV's trajectory $\{S, P_1, P_2, \ldots, P_n, E\}$. In this study, the multi-UAV path planning problem is modeled as a labeled path generation problem, where each UAV is assigned a specific origin and destination. A feasible path from the origin to the destination must be generated for each UAV while satisfying flight safety and path feasibility constraints. Considering various constraints, the multi-UAV path cost function is established as a measure for assessing path quality. The path cost function mainly encompasses fuel cost, threat cost, altitude cost, angle cos, and cooperation cost [5]. The cooperation cost includes time cost and distance cost. The objective function for collaborative multi-UAV routing is:

$$O = \omega_1 \cdot O_{fuel} + \omega_2 \cdot O_{threat} + \omega_3 \cdot O_{angle} + \omega_4 \cdot O_{height} + \omega_5 \cdot O_{time} + \omega_6 \cdot O_{dis} \tag{1}$$

where ω_1, ω_2, ω_3, ω_4, ω_5 and ω_6 represent weighting coefficients for differernt cost functions and sum to one. By adjusting the value of the weighting coefficient, the UAV flight path satisfying different requirements can be obtained [23].

2.1 Flight environment

In multi-UAV route planning, it is necessary to establish a suitable planning space according to the flight environment. In the current work, the task working environment is set to mountain, and elevation data, simulating mountain hazards, comes from a stochastically generated digital terrain model [24]. This model includes the raw digital model and the threat equivalent terrain model, where the raw data model is represented as

$$\begin{aligned} z_1(x, y) = \sin(y + a) + b \cdot \sin(x) + c \cdot \cos\left(d \cdot \sqrt{x^2 + y^2}\right) \\ + e \cdot \cos(y) + f \cdot \sin\left(f \cdot \sqrt{x^2 + y^2}\right) + g \cdot \cos(y) \end{aligned} \tag{2}$$

where x and y represent the coordinates of the horizontal projection of the original digital model; z_1 denotes elevation within the X-Y plane; a, b, c, d, e, f and g represent constants, and diverse landscapes can be achieved by adjusting these coefficient values.

The terrain model of threat equivalence is

$$z_2(x, y) = \sum_{i=1}^{K} h(i) \cdot exp\left(-\left(\frac{x - x_{0i}}{x_{si}}\right)^2 - \left(\frac{y - y_{0i}}{y_{si}}\right)^2\right) \tag{3}$$

where x and y represent the horizontal coordinates of the threat equivalent terrain projection; z_2 denotes the elevation linked to the horizontal axis in the threat equivalence framework. The height of peak i is denoted by $h(i)$, while its apex is located at the coordinates (x_{0i}, y_{0i}). The variables x_{si} and y_{si} represent parameters that influence the gradient of peak i on the X and Y axes, respectively. When x_{si} and y_{si} take on larger values, the slope of the peak becomes more gradual.

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

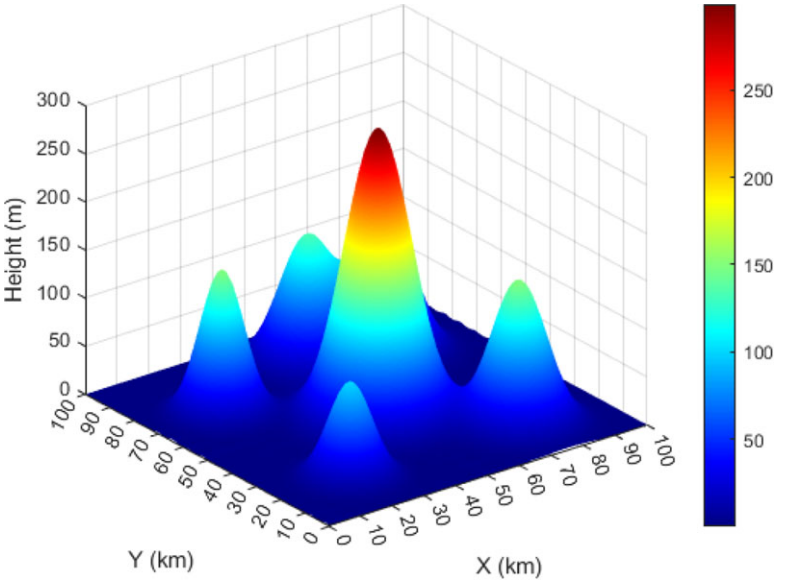


Figure 1. *The 3D terrain planning environment.*

The ultimate obstacle model merges the initial digital model with the threat-equivalent terrain model, selecting the higher value between the two.

$$z(x, y) = \max\left[z_1(x, y)\,, z_2(x, y)\right] \tag{4}$$

Various landscapes can be generated by adjusting the function's parameters. The 3D terrain planning environment is shown in Fig. 1.

2.2 Threat cost

During UAV path planning, the UAV must steer clear of obstacles. In this paper, the obstacles are the peaks, modeled as cones, and the paths formed between two adjacent waypoints of the UAV should avoid the peaks. Between every two waypoints, it is split into $m + 1$ segments to obtain m sampling points, and the threat cost between two adjacent waypoints is calculated according to m sampling points. At the same time, the height condition is set, when the UAV's altitude exceeds the mountain's peak height, the threat cost of the UAV is 0, and the UAV flight safety is guaranteed by its altitude.

The threat cost function of UAV is calculated as follows:

$$O_{threat} = \sum_{i=0}^{n} \sum_{j=0}^{m} \sum_{k=1}^{K} threat_{ij}\,(k) \tag{5}$$

$$threat_{ij}(k) = \begin{cases} 0 & \mathrm{h_j} > H(k) \ \text{ or } \Delta\mathrm{d} > \mathrm{R_T} \\ R_T - \Delta d & \mathrm{h_j} < H(k) \ \text{ and } \Delta\mathrm{d} < \mathrm{R_T} \end{cases}$$

$$\Delta d = d_T - d_{Tmin} \tag{7}$$

$$R_T = \left(H(k) - h_j\right) / tan\theta \tag{8}$$

where $H(k)$ denotes peak k's height and h_j represents the UAV's flight altitude; R_T refers to the obstacle radius corresponding to the current UAV's altitude, and θ represents the incline of the current mountain obstacle; d_T denotes the horizontal distance from the UAV to the obstacle's centre; d_{Tmin} is the minimum

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

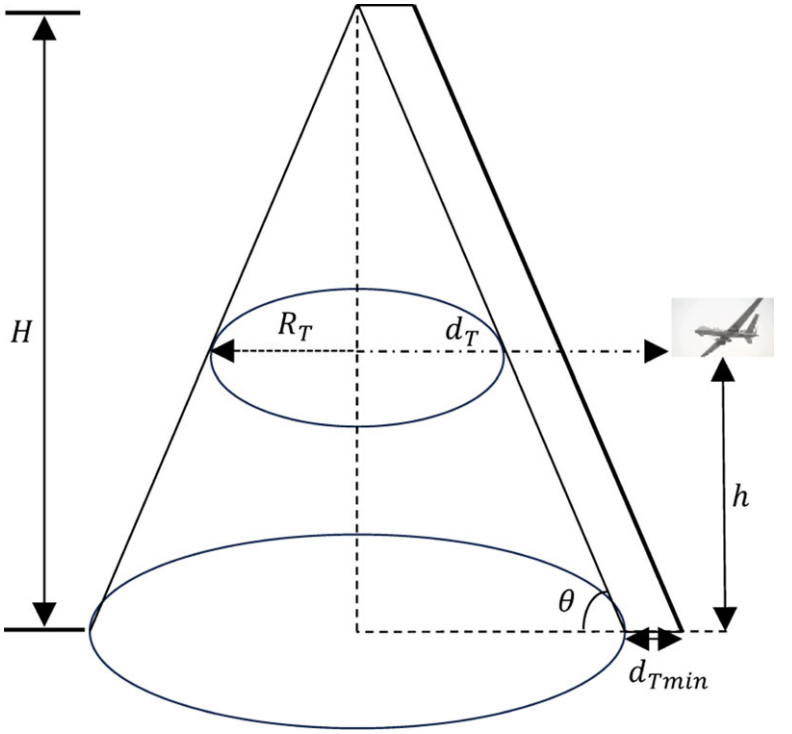


Figure 2. *Diagram of obstacle avoidance.*

distance limit between the UAV and the obstacle that is feasible for obstacle avoidance. The diagram of obstacle avoidance is shown in Fig. 2.

2.3 Fuel cost

In the UAV route optimisation challenge, fuel consumption is the primary concern. UAV flight path distance directly impacts fuel consumption. The fuel cost function is proportional to the path length. Therefore, minimising fuel consumption means minimising the path length of the UAV. As can be seen from the above, a series of N waypoints has been established between the UAV's launch and landing sites. By determining the Euclidean distance between each pair of neighbouring waypoints, we can accurately approximate the drones' overall trajectory distances. The UAV's fuel cost is determined by the formula:

$$O_{fuel} = \sum_{i=0}^{n} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \tag{9}$$

where (x_i, y_i, z_i) and $(x_{i+1}, y_{i+1}, z_{i+1})$ represent the 3D positions of the current and next waypoints, respectively. (x_0, y_0, z_0) denotes the initial position, $(x_{n+1}, y_{n+1}, z_{n+1})$ denotes the final position.

2.4 Angle cost

Angle cost is an important metric used to evaluate and optimise the smoothness of the route in UAV path planning. By calculating the angle of steering and climbing between two adjacent waypoints in the UAV path, the path can be ensured to be as smooth as possible in the path optimisation process, thus improving flight efficiency and safety. Where the steering angle is φ, the climbing angle is β.

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

The angle cost is calculated as follows:

$$O_{angle} = \sum_{i=0}^{n} (\beta_i + \varphi_i) \tag{10}$$

$$\beta = \left| \arctan\left(\frac{|z_{i+1} - z_i|}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \right) \right| \tag{11}$$

$$\varphi_i = \left| \arctan\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right) \right| \tag{12}$$

where (x_i, y_i, z_i) and $(x_{i+1}, y_{i+1}, z_{i+1})$ correspond to the 3D coordinates of the current waypoint and the next waypoint.

2.5 Altitude cost

During the flight of the UAV, it must always maintain the altitude within a certain range to ensure the safety of the UAV. The altitude cost is calculated as follows:

$$O_{height} = \sum_{i=1}^{n} \Delta h(i) \tag{13}$$

$$\Delta h(i) = \begin{cases} h_i - h_{safe_i} & \text{if } \mathrm{h_i} > \mathrm{h_{safe_i}} \\ 200 \cdot \left(h_{safe_i} - h_i \right) & \text{otherwise} \end{cases} \tag{14}$$

where h_{safe_i} represents the safety height of the ith waypoint. It can be calculated from Equations (2–4), h_{safe_i} can be expressed by the following equation:

$$h_{safe_i} = z(x_i, y_i) = \max\left[z_1(x_i, y_i) , z_2(x_i, y_i) \right] \tag{15}$$

where $z(x_i, y_i)$ represents the terrain height in the flight environment corresponding to the horizontal coordinates of the waypoint (x_i, y_i).

2.6 Cooperation cost

In this paper, the cooperation cost mainly includes the time cost as well as the multi-UAV distance cost. Temporal cooperation requires that all UAVs arrive at the destination at the same time as possible, and if a certain route does not satisfy the temporal cooperation constraint, the path should be modified. The multi-UAV distance cost requires that the distance between two UAVs must be greater than the minimum distance limit, so as to avoid the collision of multiple UAVs.

The speed of the UAVs is assumed to be within $[v_{min}, v_{max}]$ and the ith UAV's time interval is $T_i \in [T_{imin}, T_{imax}]$. Meanwhile, the flight time interval of the jth UAV is $T_j \in [T_{jmin}, T_{jmax}]$. If the flight times of two UAVs intersect, then two UAVs satisfy the temporal cooperation condition, that is,

$$T_{inter} = T_i \cap T_j \neq \emptyset \tag{16}$$

$$T_{imin} = \frac{L_i}{v_{max}} \tag{17}$$

$$T_{imax} = \frac{L_i}{v_{min}} \tag{18}$$

where L_i refers to the trajectory distance of the ith drone, here replaced by the fuel cost. We define T_{min} as the minimum time interval for the overall intersection of multiple UAVs. It is set that the ratio of the overlapping time interval of the two UAVs to the shortest interval duration is at least α, that is

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

$T_{inter} > \alpha \cdot T_{min}$, we define $0 < \alpha \leq 1$. The time cost function is as follows:

$$O_{time} = \begin{cases} 300 & \mathrm{T_{inter}} = \emptyset \\ 300 \cdot \alpha & 0 < \mathrm{T_{inter}} \leq \alpha \cdot \mathrm{T_{min}} \\ 0 & \mathrm{T_{inter}} > \alpha \cdot \mathrm{T_{min}} \end{cases} \tag{19}$$

Multi-UAVs should always pay attention to obstacle avoidance during flight; that is, the distance between any point on the path of two UAVs should be greater than the minimum distance limit. Then the multi-UAV distance cost function is shown as follows:

$$O_{dis} = \sum_{i=1}^{M} \sum_{k=1}^{n} p_{i,k} \tag{20}$$

$$p_{i,k} = \begin{cases} 10 & \text{if } \mathrm{d_{i-j,k}} < \mathrm{d_{min}} \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

where $p_{i,k}$ is the penalisation function of the spatial distance of multiple UAVs. m denotes the drone count and n indicates the path point quantity. $d_{i-j,k}$ represents distance of the kth path point on the ith and jth UAV trajectories. d_{min} represents the minimum spatial distance limit between two UAVs.

3.0 The proposed chaotic grey wolf optimiser

To address constrained optimisation during collaborative multi-UAV pathfinding, this study introduces a trajectory planning method utilising CGWO. The algorithm combines the advantages of chaotic mapping and classical GWO, and improves the algorithm's global and local optimisation performance by improving the population initialisation strategy, local chaotic search and leader optimisation.

3.1 The GWO algorithm

Mirjalili et al. introduced GWO, a swarm-based algorithm mimicking wolf predation [16]. The algorithm mimics the social ranking and predatory tactics observed in wild grey wolves and achieves the solution of complex optimisation problems through simple and efficient rules.

In the GWO, the grey wolf population is categorised into four levels: individual at the highest level is referred to as α wolf, the α wolf represents the group's best outcome and is the leader of the entire population, with the main responsibility of being in charge of assisting subjects within a cohort to forage; the suboptimal solution in the population is known as the β wolf, β wolf act as assistants to α wolf, supplementing α wolf to help manage and coordinate pack behaviour; the sub-suboptimal individual in the group is the δ wolf, whose role is to provide advice and offer decision support; individuals in the remaining populations are defined as ω wolves, following the other tier individuals as they move around, regulating internal relationships and ensuring balance within the population.

The hunting formula for wolves is:

$$D = \left| C \cdot X_p(t) - X(t) \right| \tag{22}$$

$$X(t+1) = X_p(t) - A \cdot D \tag{23}$$

$$A = 2 \cdot a \cdot r_1 - a \tag{24}$$

$$C = 2 \cdot r_2 \tag{25}$$

$$a = 2 - \frac{2 \cdot t}{T} \tag{26}$$

where t represents the iteration count; $X(t)$ denotes a wolf's present location; $X_p(t)$ denotes the prey's location vector; A and C are factors; r_1 and r_2 are arbitrary numbers in the interval [0,1]; a decreases linearly from 2 to 0.

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

The remaining grey wolf individuals in the population follow the three wolves with higher priority for position location updates. The positional update formula is:

$$D_\alpha = |C_1 \cdot X_\alpha(t) - X(t)| \tag{27}$$

$$D_\beta = \left|C_2 \cdot X_\beta(t) - X(t)\right| \tag{28}$$

$$D_\delta = |C_3 \cdot X_\delta(t) - X(t)| \tag{29}$$

$$X_1 = X_\alpha - A_1 \cdot D_\alpha \tag{30}$$

$$X_2 = X_\beta - A_2 \cdot D_\beta \tag{31}$$

$$X_3 = X_\delta - A_3 \cdot D_\delta \tag{32}$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \tag{33}$$

where X_α, X_β and X_δ refer to the location vectors for α, β and δ within the present group, correspondingly; D_α, D_β and D_δ denote distances from a grey wolf candidate to the top three wolves [22].

3.2 Algorithm improvements and enhancements

GWO is a smart algorithm inspired by grey wolf social dynamics, which exhibits structural simplicity and robust global exploration capabilities. However, GWO may have problems like slow convergence and a propensity to become stuck in local optima when facing complex optimisation challenges. In contrast, chaotic sequences are randomised and globally traversable, which can improve the search capability by providing comprehensive coverage in a limited search space. Therefore, chaotic sequences can be combined with the GWO algorithm to avert GWO's entrapment in local minima and sharpen convergence precision [25].

The logistic map, represented by an equation, exemplifies a standard 1D chaotic system.

$$x_{k+1} = \mu \cdot x_k \cdot (1 - x_k) \tag{34}$$

where μ represents control parameter, x_k represents relative number in the kth generation, the initial value x_0 takes any value from 0 to 1. different x_0 and μ can generate different sequences, and when μ is equal to 4, a completely chaotic sequence is obtained [26]. According to this property, the GWO algorithm can escape from local optima.

Meanwhile, GWO's iterative process exhibits issues with early convergence and limited adaptability, so the quality of global search capability and reconciliation is enhanced by dynamically adjusting the leadership during GWO iterations.

3.2.1 Chaos initialisation

The traditional GWO usually uses a stochastic approach for initialisation, randomly generating the initial position of each individual in a predefined search space and then starting subsequent iterations. The random initialisation formula is as follows:

$$P_k = lb + r \cdot (ub - lb)\,, k = 1, 2, \cdots, N-1, N \tag{35}$$

where P_k denotes the initial position of the individuals within the population; r is a stochastic variable within the interval [0,1]; ub represents the search space's upper limit, and lb denotes its lower limit.

In chaotic initialisation, the random value x_k generated by Equation (34] is used instead of r in Equation (35], and the positions of $2N$ individuals are initialised at the same time, which are sorted by the fitness function, and the top N individuals with minimal fitness define the population's initial positions. the chaos initialisation formula is:

$$P_k = lb + x_k \cdot (ub - lb)\,, k = 1, 2, \cdots, 2N-1, 2N \tag{36}$$

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

3.2.2 Chaotic local search

Chaotic local search combines logic mapping and optimisation algorithms for local search, and the basic idea is to use the traversal and randomness of chaotic sequences to generate random solutions in the solution space, thus improving the ability and efficiency of local search.

In this improved algorithm, in each iteration, the optimal individual α wolf is first sorted according to the fitness function to find out the optimal individual α wolf, and then according to the logical mapping Equation (34), a local search is performed around the optimal individual to obtain a new individual grey wolf. The logical local search formula is:

$$X_c = X_\alpha + R \cdot (z - 0.5) \tag{37}$$

where z refers to chaotic random sequences generated by k iterations using logistic mapping Equation (34], $z = \{x_0, x_1, \cdots, x_n\}$; R denotes the local search scope influencing neighbourhood exploration, while X_α signifies the α wolf's location.

To determine the fitness value of individual X_c using the fitness function, assess how it stacks up against the current optimal individual in this iteration of the GWO algorithm. If X_c's fitness value turns out to be lower than that of the existing optimal individual, it should take the reins as the new α wolf. On the other hand, if X_c doesn't outperform the current leader, the α wolf retains its position without any changes.

3.2.3. Dynamic leadership downsizing mechanism

The leader dynamic reduction mechanism, as an improvement mechanism in GWO, aims to gradually reduce the size of the leader through iterations to enhance the algorithm's final-stage local search performance, and concurrently boost convergence speed [22].

In the improved algorithm, γ wolf is introduced, which is less adaptive than δ wolf in terms of fitness, and the related formula is as follows:

$$D_\gamma = \left| C_4 \cdot X_\gamma(t) - X(t) \right| \tag{38}$$

$$X_4 = X_\gamma - A_4 \cdot D_\gamma \tag{39}$$

The specific leadership downsizing mechanism formula is as follows:

$$X(t+1) = \begin{cases} \frac{X_1+X_2+X_3+X_4}{4} & 0 \le \mathrm{t} \le \frac{\mathrm{T}}{3} \\ \frac{X_1+X_2+X_3}{3} & \frac{\mathrm{T}}{3} < t \le \frac{2\mathrm{T}}{3} \\ \frac{X_1+X_2}{2} & \frac{2\mathrm{T}}{3} < t \le T \end{cases} \tag{40}$$

The dynamic reduction mechanism of the leadership layer by gradually reducing the size of the leadership layer, using the synergy of the four leadership layers for extensive global exploration and increasing population diversity during initial optimisation, and focusing on two leadership layers at the late stage of optimisation to enhance local process accuracy and speed of convergence, this mechanism enhances the optimisation algorithm's performance by balancing exploration and exploitation abilities.

3.3 CGWO algorithm code

The chaos grey wolf optimiser pseudo-code is shown below in algorithm 1.

4.0 Algorithm performance testing

4.1 Benchmark function

In this section, we comprehensively tested the overall performance of the CGWO algorithm by using the internationally widely recognised CEC2017 test set [27]. The CEC2017 benchmark test set comprises 29 evaluation functions, and the specific information is shown in Table 1. Where F_1 and F_3 are unimodal; $F_4 - F_{10}$ are basic multimodal; $F_{11} - F_{20}$ are hybrid; and $F_{21} - F_{30}$ are composite functions.

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

Algorithm 1 CGWO algorithm

1: Grey wolf population chaos initialisation;
2: Calculate fitness values and choose the highest-ranked n individuals;
3: Find the first four individuals as X_α, X_β, X_δ and X_γ;
4: Initialise parameters a, A and C according to Equations (24–26);
5: **while** $t \leq T$ **do**
6: Each individual updates its position according to Equation (40);
7: Update a, A and C;
8: Calculate fitness value of each individuals according to Equation (2) and find the optimal individual X_α;
9: Perform a chaotic local search according to Equation (37) around the optimal individual and update X_α;
10: $t = t + 1$;
11: **end while**
12: Output the position of α wolf

The CEC2017 benchmark test set validates the performance of the optimization algorithms in a number of different situations and dimensions by designing a series of challenging optimisation problems. It assesses both algorithms' broad exploratory power and its adaptability to restrictions and intricate settings.

4.2 Parameter setting

We selected four representative algorithms to compare their performance with the CGWO, namely the PSO algorithm [12], the whale optimisation algorithm (WOA) [28], the artificial ecosystem optimisation (AEO) [29], and the GWO [16].

PSO is a classical algorithm that guides the search process through the interaction of the global optimal solution and the local optimal solution to achieve the optimal solution of the problem; WOA mirrors the foraging patterns of humpbacks; this strategy diminishes premature convergence and boosts comprehensive exploration; AEO models organismal interactions within ecosystems, proving efficient for intricate, multi-parameter optimisation; GWO, in which the α wolf guides the other wolves and eventually approaches the optimal solution, converges rapidly but easily gets trapped in local optima. Table 2 displays the configuration parameters for each algorithm; to ensure fairness, both GWO and CGWO have the same common parameter settings.

To minimise total experimental error, the parameters are set uniformly throughout the experiment. Each function in the CEC2017 test set is simulated independently for all algorithms for 10 times, and each iteration is set to have a maximum number of iterations of 300, with a test dimension of 30 dimensions, while all algorithm population sizes are set to 30.

4.3 Performance analysis

This section aims to analyse the performance of CGWO and the four algorithms it compares with. It is divided into two parts. The first part analyses the quality of algorithm optimisation by running the above algorithms on the CEC2017 test set and obtaining the average and standard deviation of the optimal solutions for each algorithm. The second part conducts a detailed analysis of the convergence performance of the algorithms by plotting their convergence curves in CEC2017.

4.3.1 Optimisation quality analysis

Tables 3 and 4 summarise the results of the five algorithms listed above on the CEC2017 test set. Mean refers to the average of the optimal values obtained by running each algorithms independently 10 times,

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

Table 1. *Detailed information of CEC2017*

Function Number	Function Name	Search Range	Theoretical Optimum
F_1	Shifted and rotated bent cigar function	[−100,100]	100
F_3	Shifted and rotated Zakharov function	[−100,100]	300
F_4	Shifted and rotated Rosenbrock's function	[−100,100]	400
F_5	Shifted and rotated Rastrigin's function	[−100,100]	500
F_6	Shifted and rotated expanded Scaffer's F6 function	[−100,100]	600
F_7	Shifted and rotated Lunacek Bi-Rastrigin function	[−100,100]	700
F_8	Shifted and rotated noncontinuous Rastrigin's function	[−100,100]	800
F_9	Shifted and rotated Levy function	[−100,100]	900
F_{10}	Shifted and rotated Schwefel's function	[−100,100]	1000
F_{11}	Hybrid function 1 ($N=3$)	[−100,100]	1100
F_{12}	Hybrid function 2 ($N=3$)	[−100,100]	1200
F_{13}	Hybrid function 3 ($N=3$)	[−100,100]	1300
F_{14}	Hybrid function 4 ($N=4$)	[−100,100]	1400
F_{15}	Hybrid function 5 ($N=4$)	[−100,100]	1500
F_{16}	Hybrid function 6 ($N=4$)	[−100,100]	1600
F_{17}	Hybrid function 6 ($N=5$)	[−100,100]	1700
F_{18}	Hybrid function 6 ($N=5$)	[−100,100]	1800
F_{19}	Hybrid function 6 ($N=5$)	[−100,100]	1900
F_{20}	Hybrid function 6 ($N=6$)	[−100,100]	2000
F_{21}	Composition function 1 ($N=3$)	[−100,100]	2100
F_{22}	Composition function 2 ($N=3$)	[−100,100]	2200
F_{23}	Composition function 3 ($N=4$)	[−100,100]	2300
F_{24}	Composition function 4 ($N=4$)	[−100,100]	2400
F_{25}	Composition function 5 ($N=5$)	[−100,100]	2500
F_{26}	Composition function 6 ($N=5$)	[−100,100]	2600
F_{27}	Composition function 7 ($N=6$)	[−100,100]	2700
F_{28}	Composition function 8 ($N=6$)	[−100,100]	2800
F_{29}	Composition function 9 ($N=3$)	[−100,100]	2900
F_{30}	Composition function 10 ($N=3$)	[−100,100]	3000

and Std refers to the standard deviation of the optimal values obtained by each algorithm. The best results in the chart are displayed in bold.

The experimental results in Tables 3 and 4 show that CGWO has strong versatility in optimising different types of problems. Among a total of 29 test functions, CGWO obtained results close to the theoretical optimum in 20 functions, and its results consistently ranked in the top two among all test functions, demonstrating excellent optimisation quality. In the nine test functions F_1, F_4, F_{12}, F_{13}, F_{14}, F_{15}, F_{18}, F_{19} and F_{30}, the solution accuracy of CGWO is significantly higher than that of other algorithms, with an average difference of one order of magnitude. In particular, for F_1 and F_{13}, the average value of the optimal solution obtained by CGWO is three orders of magnitude smaller than that of other

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

Table 2. *Parameter values of algorithms*

Algorithm	Parameter	Value
PSO	C_1, C_2	1.5
	ω	[0.4, 0.9]
	V_{max}	0.5
	V_{min}	−0.5
	r_1, r_2	[0, 1]
WOA	a	$2 \rightarrow 0$
	A	$[-a, a]$
	C	[0, 2]
	r	[0, 1]
	b	1
	l	[−1, 1]
AEO	r	[0, 1]
	v_1, v_2	$N(0, 1)$
	C	$\frac{v_1}{2\lvert v_2\rvert}$
	u	$N(0, 1)$
	D	u 3
	h	[−1, 1]
GWO	r_1, r_2	[0, 1]
	a	$2 \rightarrow 0$
	A	$[-a, a]$
	C	[0, 2]
CGWO	r_1, r_2	[0, 1]
	a	$2 \rightarrow 0$
	A	$[-a, a]$
	C	[0, 2]
	μ	4
	R	1.7

algorithms. For the aforementioned nine test functions, CGWO achieves the smallest standard deviation after running independently 10 times. By comparing the average values and standard deviations of the optimal values across all algorithms, it can be concluded that CGWO demonstrates outstanding optimisation performance. The significant superiority of CGWO indicates that the chaotic logic mapping method proposed in this paper enhances population diversity, thereby increasing the likelihood of the algorithm finding high-quality solutions.

4.3.2 Convergence analysis

Figure 3 offers a visual depiction of algorithmic convergence on the CEC2017 benchmark test. In general, the CGWO algorithm has optimal convergence performance. At the beginning of the population iteration, CGWO converges significantly faster than other algorithms, which demonstrates the CGWO algorithm's robust search capacity and efficient strategy. During the mid-to-late phases of the population iteration, the fitness value of CGWO is smaller than the other algorithms, demonstrating that CGWO converges earlier and identifies the global optimum more rapidly than other algorithms.

For the test functions F_1, F_8, F_{12}, F_{13}, F_{18},F_{20}, F_{22} and F_{30}, the CGWO algorithm shows the best convergence performance and is substantially ahead of the other algorithms, discovering smaller global optima compared to alternative algorithms in 300 generations. For the test functions F_4, F_7, F_{11}, F_{14}, F_{17}, F_{19}, F_{25} and F_{28}, the CGWO algorithm does not manage to achieve a significant lead like the previous test functions, but still achieves the smallest fitness value, which also proves the advanced nature of the CGWO algorithm.

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

Table 3. *CEC2017 test results (F1–F15)*

Function	Performance	PSO	WOA	AEO	GWO	CGWO
F_1	Mean	3.68193e+10	1.04712e+10	1.98761e+10	3.84315e+09	**6.34341e+06**
	Std	4.92627e+09	2.16091e+09	5.50047e+09	1.83151e+09	**2.54874e+06**
F_3	Mean	6.53514e+04	2.73500e+05	8.86826e+04	8.65247e+04	**2.00040e+04**
	Std	1.15703e+04	8.55243e+04	1.34462e+04	2.73834e+04	**8.26120e+03**
F_4	Mean	1.06706e+04	1.77985e+03	2.44529e+03	6.87402e+02	**4.67470e+02**
	Std	3.42494e+03	5.20932e+02	1.20217e+03	1.09076e+02	**3.60273e+01**
F_5	Mean	8.89955e+02	8.53060e+02	8.24034e+02	**6.42350e+02**	6.96135e+02
	Std	**3.30180e+01**	6.43301e+01	3.81429e+01	4.70085e+01	6.63613e+01
F_6	Mean	6.89022e+02	6.87568e+02	6.58087e+02	**6.14659e+02**	6.35534e+02
	Std	**4.15918e+00**	1.56727e+01	5.34459e+00	5.99575e+00	1.18450e+01
F_7	Mean	1.36707e+03	1.34751e+03	1.30577e+03	**9.45688e+02**	9.84853e+02
	Std	6.48140e+01	5.80060e+01	8.47371e+01	7.42092e+01	**2.88047e+01**
F_8	Mean	1.10098e+03	1.09261e+03	1.08346e+03	**9.48659e+02**	9.96701e+02
	Std	**2.87223e+01**	3.98974e+01	3.23930e+01	5.24702e+01	5.37444e+01
F_9	Mean	7.82690e+03	1.36055e+04	7.89487e+03	**2.45397e+03**	3.81012e+03
	Std	1.07329e+03	5.15531e+03	1.51550e+03	**7.40062e+02**	1.02091e+03
F_{10}	Mean	8.33236e+03	7.67545e+03	8.67476e+03	6.11667e+03	**5.92109e+03**
	Std	4.35573e+02	8.15234e+02	**2.45062e+02**	1.99916e+03	5.16349e+02
F_{11}	Mean	6.56836e+03	1.33136e+04	4.33790e+03	3.06873e+03	**1.32606e+03**
	Std	1.63494e+03	5.92299e+03	1.30572e+03	1.81629e+03	**4.47826e+01**
F_{12}	Mean	8.37240e+09	8.71979e+08	1.43682e+09	1.01056e+08	**2.04193e+07**
	Std	3.87410e+09	5.39342e+08	6.18787e+08	1.19078e+08	**1.04370e+07**
F_{13}	Mean	2.11241e+09	7.38810e+07	4.95098e+08	5.80329e+07	**3.05970e+05**
	Std	1.76689e+09	1.03512e+08	5.43203e+08	1.13917e+08	**1.16700e+05**
F_{14}	Mean	2.08228e+05	2.88301e+06	6.64602e+05	9.14959e+05	**4.12209e+04**
	Std	3.19379e+05	5.54509e+06	4.31478e+05	9.47320e+05	**3.64996e+04**
F_{15}	Mean	8.31695e+05	1.47309e+07	3.35883e+07	1.40450e+06	**9.69079e+04**
	Std	2.54879e+06	1.69703e+07	3.43497e+07	2.81775e+06	**5.73971e+04**

The experimental results presented in this chapter demonstrate the key role of chaotic initialisation, chaotic local search and leader dynamic reduction mechanisms in enhancing the efficiency of the GWO algorithm. It is proved that chaotic initialisation demonstrably yields a more uniformly distributed and randomised initial population, significantly enhancing the GWO algorithm's initial search efficiency, making it more diversified and capable of global exploration; chaotic localised search prevents the issue of traditional methods being trapped in local optima due to over-reliance on the current optimal solution, and enhances search varity and acclerates convergence. The dynamic reduction of the leader provides an adaptive balance between global and local search, which improves the convergence speed and accuracy of the algorithm. The combination of the three can effectively improve the convergence performance of GWO in complex problems, avoid premature convergence and improve the overall performance.

5.0 3D Multi-uav path planning with cgwo algorithm

This section presents two drone deployment scenarios illustrating CGWO's efficacy in practical applications.

5.1 Experimental parameters

In the experiment, the overall flight space is set to 100 km ×100 km ×500 m, which includes six peaks, and the original digital terrain model is constructed according to Equation (2]. According to Reference

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

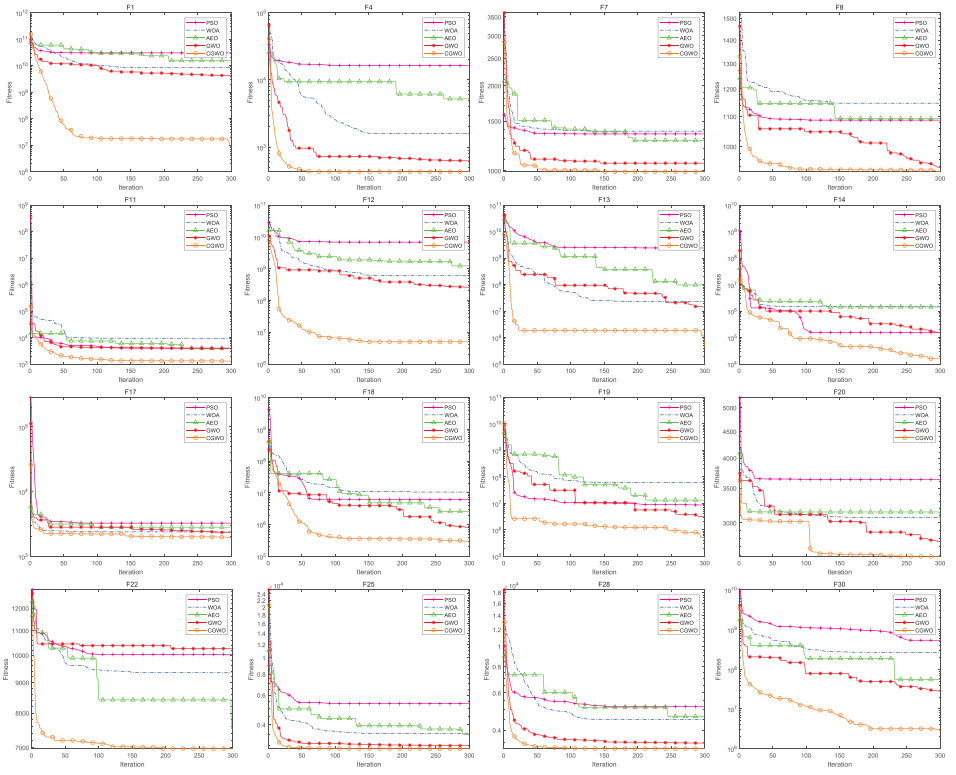


Figure 3. *Convergence performance of algorithms for different functions in CEC2017.*

[30], the initial terrain model parameters are configured as follows: $a = 0.1$, $b = 0.01$, $c = 1$, $d = 0.1$, $e = 0.2$, $f = 0.4$ and $g = 0.02$. The peak heights, horizontal coordinates of their apexes and the slope parameters are detailed in Table 5.

In the simulation test, we set the number of population to 120, the number of iterations to 300 and the number of path points between the start and end points to 10. The values of the weighting coefficients of the different cost functions in the total cost function are set: $\omega_1 = 0.3$, $\omega_2 = 0.25$, $\omega_3 = 0.1$, $\omega_4 = 0.1$, $\omega_5 = 0.1$ and $\omega_6 = 0.15$. The drone's speed is set at 40–60 m/s.

5.2 Case simulation

Case 1: In this case, we set up three UAVs for path planning. The starting and ending point coordinates for the three UAVs are presented in Table 6.

For Case 1, after 300 iterations, Figs. 4a and b represent the composite and top perspectives of the three UAVs' path planning, respectively, demonstrating that CGWO avoids falling into a local optima; meanwhile, it is able to maintain the distances between multiple UAVs as well as the distances between the UAVs and obstacles to avoid collisions. Figure 5 represents the cost convergence trajectories of the three UAVs. It is evident that all the curves converge quickly, validating the algorithm performance.

Case 2: In this case, we set up six UAVs for path planning. Table 7 lists starting and ending coordinates for the six UAVs.

For Case 2, after 300 iterations, Figs 6a and b represent the composite and top perspectives of the six UAVs' path planning, and according to the pictures, the distances among the six UAVs and between the UAVs and mountain peak obstacles are effectively managed, ensuring successful obstacle avoidance.

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

Table 4. *CEC2017 test results (F16–F30)*

Function	Performance	PSO	WOA	AEO	GWO	CGWO
F_{16}	Mean	5.80754e+03	4.64278e+03	4.00473e+03	2.89535e+03	**2.81439e+03**
	Std	1.26151e+03	6.15197e+02	**2.97248e+02**	4.89678e+02	5.11582e+02
F_{17}	Mean	3.66625e+03	2.96736e+03	2.67394e+03	2.17727e+03	**2.13469e+03**
	Std	5.99099e+02	4.28543e+02	2.10101e+02	**1.90487e+02**	1.90990e+02
F_{18}	Mean	3.43131e+06	2.53994e+07	6.44588e+06	7.96191e+06	**5.53871e+05**
	Std	6.00631e+06	2.85583e+07	4.31025e+06	8.88596e+06	**2.60098e+05**
F_{19}	Mean	1.23647e+07	2.35067e+07	3.33161e+07	1.80531e+06	**6.84978e+05**
	Std	2.23805e+07	1.66475e+07	3.21291e+07	1.71620e+06	**6.14432e+05**
F_{20}	Mean	3.11264e+03	3.07800e+03	2.92073e+03	2.52090e+03	**2.50288e+03**
	Std	2.45843e+02	1.88839e+02	**1.30020e+02**	3.15260e+02	1.62626e+02
F_{21}	Mean	2.79786e+03	2.67716e+03	2.59505e+03	**2.42417e+03**	2.52765e+03
	Std	7.28979e+01	4.18853e+01	**1.55639e+01**	4.65971e+01	2.94861e+01
F_{22}	Mean	9.60408e+03	8.58445e+03	4.48331e+03	5.23472e+03	**3.80101e+03**
	Std	3.61280e+02	1.71765e+03	7.77104e+02	**1.68722e+03**	2.40676e+03
F_{23}	Mean	4.23734e+03	3.17128e+03	3.11253e+03	**2.81273e+03**	2.91556e+03
	Std	9.38393e+02	1.09503e+02	6.77687e+01	7.44143e+01	**6.15554e+01**
F_{24}	Mean	4.09371e+03	3.39048e+03	3.28124e+03	**3.02215e+03**	3.05203e+03
	Std	5.15744e+02	9.23963e+01	6.34980e+01	8.26064e+01	**6.33222e+01**
F_{25}	Mean	4.52257e+03	3.36372e+03	3.70427e+03	3.04274e+03	**2.90432e+03**
	Std	5.65870e+02	1.28445e+02	4.28714e+02	1.00021e+02	**1.97862e+01**
F_{26}	Mean	1.09369e+04	8.85369e+03	8.03180e+03	**5.04080e+03**	5.96609e+03
	Std	7.58917e+02	1.30424e+03	8.77198e+02	**2.16851e+02**	1.80265e+03
F_{27}	Mean	6.01562e+03	3.50749e+03	3.65410e+03	**3.26552e+03**	3.36254e+03
	Std	2.39237e+03	1.34543e+02	1.16316e+02	**3.56938e+01**	1.04963e+02
F_{28}	Mean	5.77083e+03	4.23963e+03	4.47122e+03	3.51365e+03	**3.26256e+03**
	Std	9.74131e+02	4.71020e+02	3.61269e+02	1.48617e+02	**1.64863e+01**
F_{29}	Mean	7.38412e+03	5.01218e+03	5.16249e+03	4.16821e+03	**4.14481e+03**
	Std	9.75322e+02	5.10151e+02	3.36907e+02	**2.25469e+02**	2.73509e+02
F_{30}	Mean	2.08611e+08	7.01447e+07	5.07512e+07	1.31632e+07	**3.97557e+06**
	Std	2.76679e+08	5.03958e+07	2.96166e+07	1.21487e+07	**2.49882e+06**

Table 5. *Threat equivalent terrain model parameters*

Number	Height	Horizontal coordinate	Slope
1	130	(56,82)	(10,10)
2	150	(75,20)	(10,10)
3	300	(50,45)	(12,12)
4	100	(22,20)	(8,8)
5	150	(20,70)	(8,8)
6	150	(75,75)	(10,10)

Table 6. *Start and end coordinates of three drones*

Number	Start point	End point
UAV1	(3,4,0)	(96,95,70)
UAV2	(14,1,0)	(98,92,70)
UAV3	(1,14,0)	(91,98,70)

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

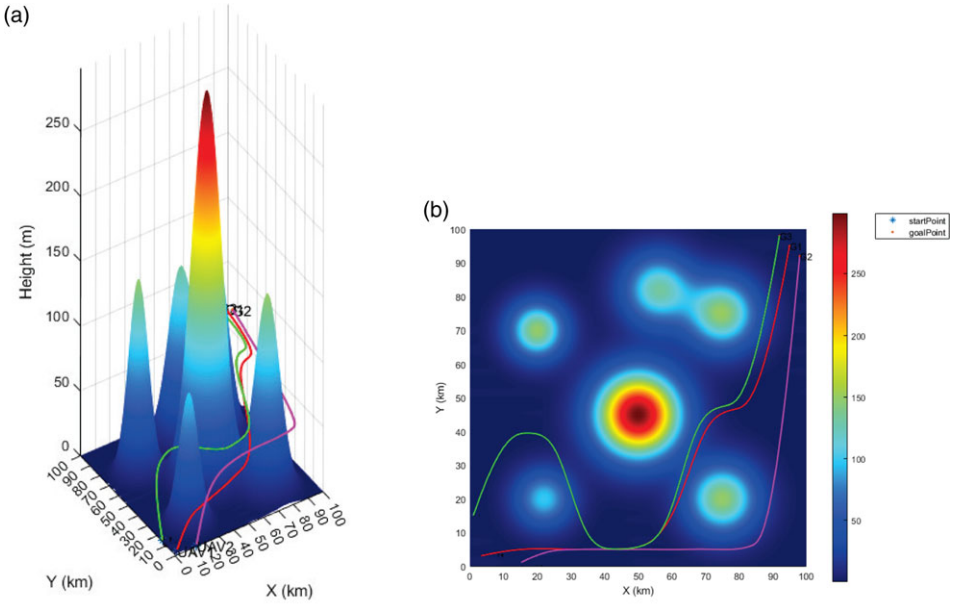


Figure 4. *Three-dimensional composite and top views of Case 1.*

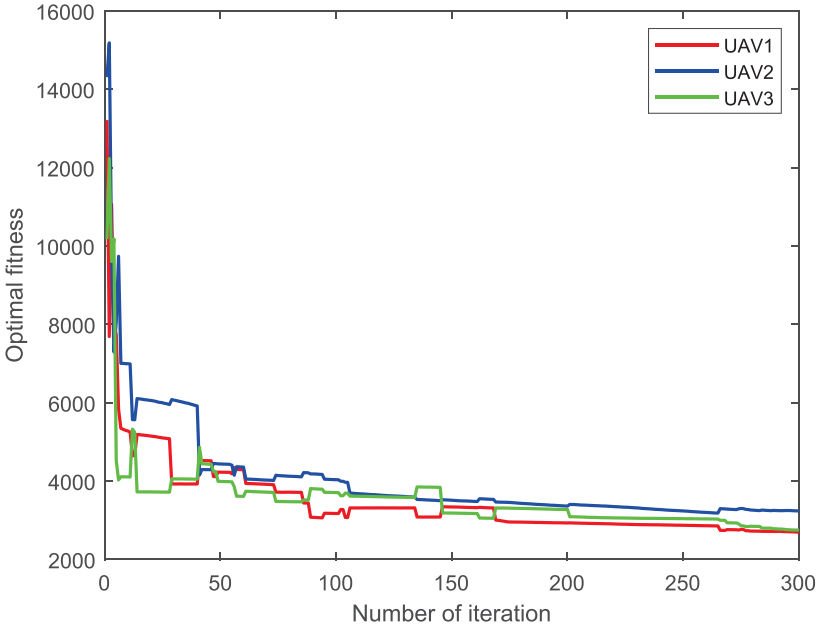


Figure 5. *Cost convergence curves for Case 1.*

Figure 7 shows the cost convergence function of the six UAVs, and all the curves converge quickly, proving the feasibility of the algorithm.

5.3 Results analysis

As shown in Figs 4 and 6, CGWO can generate flight trajectories that satisfy the constraints in both Case 1 and Case 2. Taking Case 1 as an example, CGWO and the other four comparison algorithms

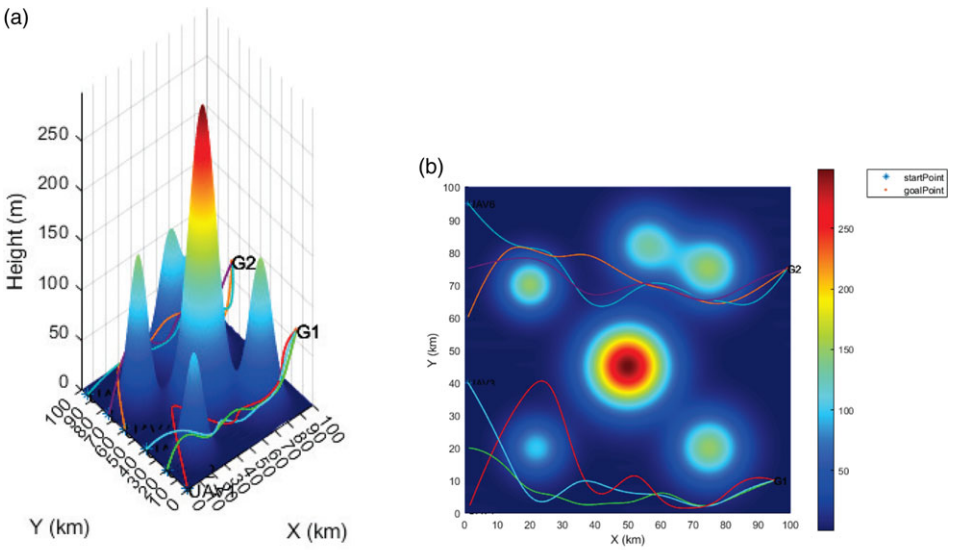
https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

Table 7. *Start and end coordinates of six drones*

Number	Start point	End point
UAV1	(1,1,0)	(97,10,70)
UAV2	(1,20,0)	(97,10,70)
UAV3	(1,40,0)	(97,10,70)
UAV4	(1,60,0)	(99,75,70)
UAV5	(1,76,0)	(99,75,70)
UAV6	(1,96,0)	(99,75,70)

Table 8. *Comparison of algorithm results*

Algorithm	Mean	Std	Best value	Worst value
CGWO	1658.16	52.63	1562.34	1753.19
GWO	1719.98	203.89	1528.98	2252.57
PSO	2022.81	44.41	1967.99	2085.99
AEO	5336.19	1317.13	3470.22	8247.83
WOA	2002.74	52.27	2063.23	1928.39



Three-dimensional composite view of Case 2 Three-dimensional top view of Case 2

Figure 6. *Three-dimensional composite and top views of Case 2.*

were simulated in the Case 1 scenario. Table 8 shows the average values and deviations of the cost function obtained by running the five algorithms 10 times in Case 1. CGWO obtains a mean value for the cost function that is smaller than the other four comparison functions. Although CGWO does not obtain an optimal value as good as GWO, its standard deviation is much smaller than that of GWO. This further proves the superiority of the proposed algorithm over the other four algorithms.

In addition, based on Case 1, the performance parameters given in reference [5, 31] were selected to verify the superiority of the proposed algorithm. Table 9 lists the parameter values mentioned in the two references. Given the different ways in which the multi-UAV path cost functions are constructed, some of the performance parameters in this paper do not exist in these references. For performance parameters that do not exist in these references, the same performance parameter settings as in this paper are used.

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

Table 9. *Performance parameters of relevant references*

Reference	Parameter	Value
5	dTmin	0
	dmin	0.02
	Vmax	60
	Vmin	40
31	dmin	0.1
	Vmax	200
	Vmin	100

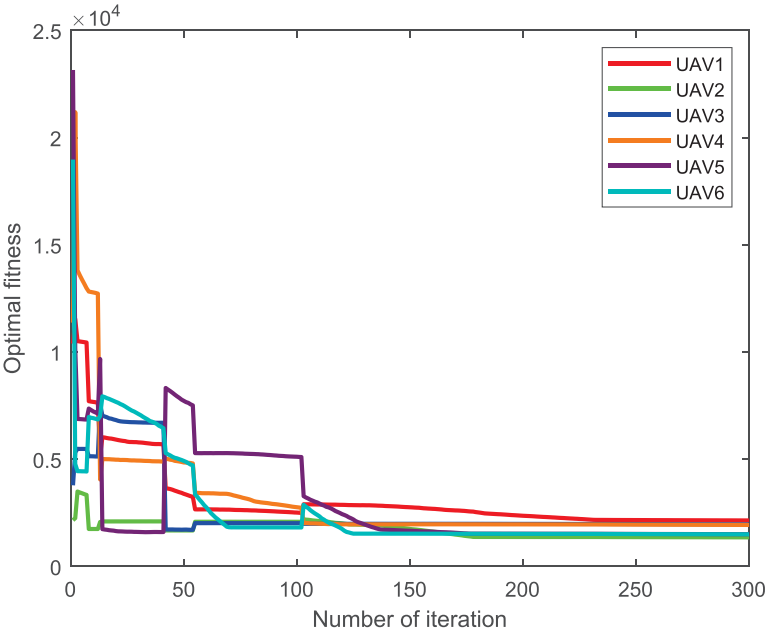


Figure 7. *Cost convergence curves for Case 2.*

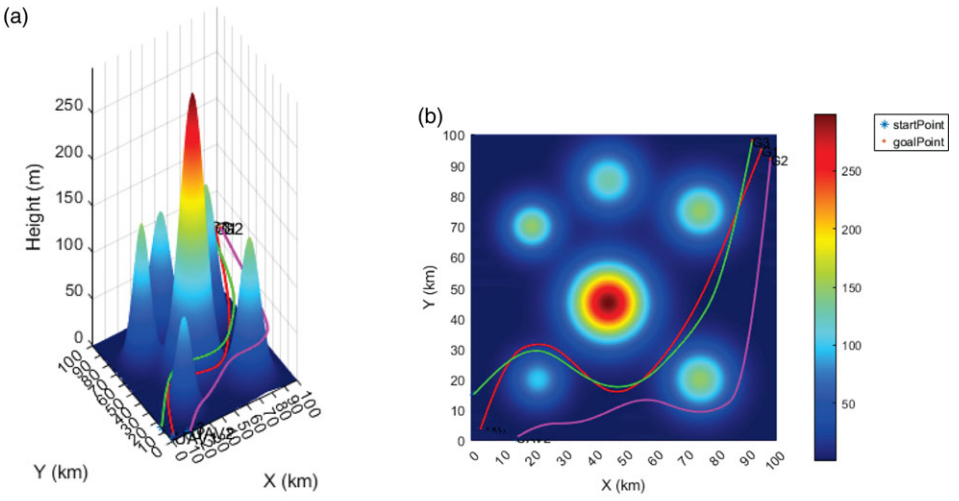
Figures 8 and 9 show the path maps obtained by simulation using the performance parameters in References [5] and [31] in Case 1. Tables 10 and 11 show the average values, standard deviations, optimal values and worst values of the cost function obtained by the proposed algorithm and the comparison algorithm under two different performance parameters.

Based on the path curves and computational results obtained from the performance parameters in the aforementioned applications and references, it can be concluded that CGWO can generate path curves that meet the conditions and are safe and collision-free under different performance parameter conditions, demonstrating excellent generalisation capabilities. Based on the computational results, it can be concluded that CGWO achieves the lowest average value of the cost function. Although CGWO ranks third in terms of standard deviation, its optimal performance in the average value of the cost function indicates that CGWO maintains global optimisation capabilities while also exhibiting stability and robustness. This further demonstrates that CGWO exhibits superior optimisation performance compared to the four comparison algorithms.

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

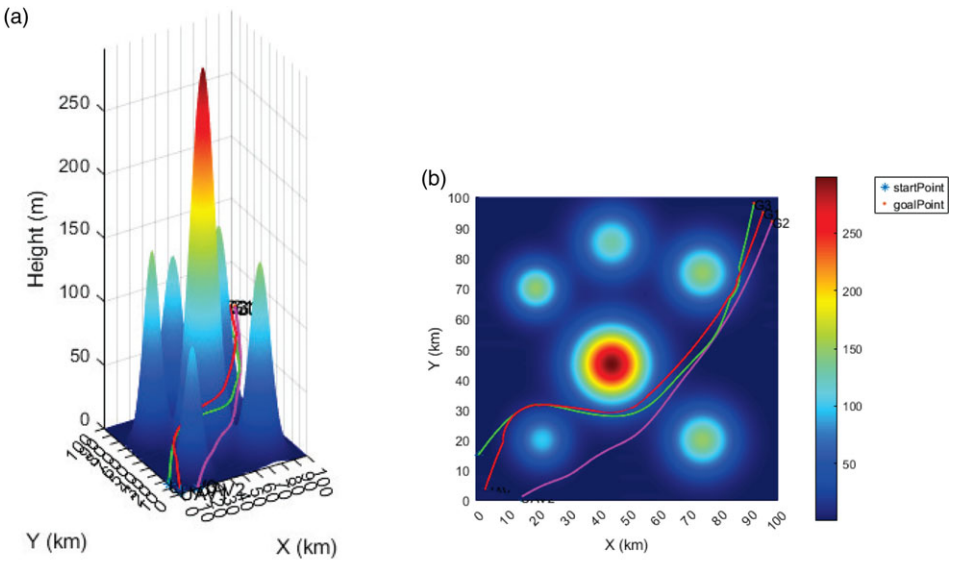
Table 10. *Algorithm results of Reference [5]*

Algorithm	Mean	Std	Best value	Worst value
CGWO	1761.03	253.67	1535.91	2252.80
GWO	1903.12	255.58	1570.63	2276.28
PSO	1820.86	6.39	1810.15	1829.91
AEO	5561.27	945.87	3748.99	6642.25
WOA	1806.87	15.77	1785.39	1838.35



Three-dimensional composite view of ref 5 Three-dimensional top view of ref 5

Figure 8. *Three-dimensional composite and top views of Reference [5].*



Three-dimensional composite view of ref 31 Three-dimensional top view of ref 31

Figure 9. *Three-dimensional composite and top views of Reference [31].*

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

Table 11. *Algorithm results of Reference [31]*

Algorithm	Mean	Std	Best value	Worst value
CGWO	1605.81	87.63	1528.49	1752.54
GWO	1721.76	311.96	1537.38	2592.34
PSO	1843.30	6.58	1833.35	1853.08
AEO	5250.33	924.97	3501.45	7038.20
WOA	1830.04	16.11	1810.10	1862.80

6.0 Conclusions

In this study, we introduce a multi-UAV trajectory design approach named as CGWO, aiming to address multi-UAV system path planning challenges. The improved algorithm builds upon the GWO framework, integrating chaotic logic mapping and a leadership-based dynamic reduction strategy. Chaotic logic mapping is applied to population initialisation and local search components. Chaotic initialisation significantly enhances population diversity, effectively expanding the coverage of the search space. To address the issue of GWO easily getting stuck in local optima, chaotic local search is introduced after individual position updates. Since chaotic logic mapping possesses universality, unpredictability, and non-periodicity, it can effectively guide individuals out of local optimum traps, thereby improving the search quality of the algorithm. The leadership dynamic decay strategy dynamically adjusts the search strategy based on the progression of the iterative process. In the early stages, the focus is on global search, gradually transitioning to local search in later stages.

In the algorithm performance test, the CEC2017 test set was used as the test basis, and numerical experiments were conducted on CGWO and four commonly used algorithms. In some test functions, the average test results obtained by CGWO are two orders of magnitude or more smaller than the average test results obtained by other algorithms. According to the numerical results obtained, it can be proven that CGWO has the best convergence accuracy in more than 66% of the test functions. According to the convergence curve diagram, the convergence speed of the CGWO algorithm is superior to that of the comparison algorithms in most cases.

In simulation experiments, the CGWO algorithm was used to generate safe paths that avoid collisions between drones and between drones and obstacles, meeting mission requirements. This greatly improved the collaborative efficiency of multi-drone systems. The comparison results also show that the path cost generated by CGWO is significantly lower than that of the other four comparison algorithms, verifying the effectiveness of this algorithm in solving multi-UAV cooperative path planning problems.

Financial support. The work was supported in part by the National Natural Science Foundation of China under Grant 62573175, in part by the Natural Science Foundation of Hebei Province under Grant F2025202029, and in part by the Natural Science Foundation of Tianjin under Grant 23JCYBJC00930.

References

[1] Arik, S., Turkmen, I. and Oktay, T. Redesign of morphing UAV for simultaneous improvement of directional stability and maximum lift/drag ratio, *Adv. Electr. Comput. Eng.*, 2018, **18**, pp 57–62.

[2] Oktay, T., and Coban, S. Simultaneous longitudinal and lateral flight control systems design for both passive and active morphing TUAVs, *Elektronika ir elektrotechnika*, 2017, **23**, pp 15–20.

[3] Yesilbas, E., Özgür, B., Ozen, E. and Oktay, T. Simultaneous and stochastic design of piston-prop TUAV vertical tail and its autonomous system, *Aircr Eng. Aerosp. Technol.*, 2025, **97**, pp 395–404.

[4] Oktay, T., Arik, S., Turkmen, I., Uzun, M. and Celik, H. Neural network based redesign of morphing UAV for simultaneous improvement of roll stability and maximum lift/drag ratio, *Aircraft Eng. Aerosp. Technol.*, 2018, **90**, pp 1203–1212.

[5] Xu, L., Cao, X., Du, W. and Li, Y. Cooperative path planning optimization for multiple UAVs with communication constraints, *Knowl-Based Syst.*, 2023, **260**, p 110164.

[6] Shen, Q., Zhang, D., He, Q., Ban, Y. and Zuo, F. A novel multi-objective dung beetle optimizer for Multi-UAV cooperative path planning, *Heliyon*, 2024, **10**, p e37286.

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press

[7] Kose, O. and Oktay, T. Simultaneous design of morphing hexarotor and autopilot system by using deep neural network and SPSA, *Aircr Eng. Aerosp. Technol*, 2023, **95**, pp 939–949.
[8] Uzun, M. and Oktay, T. Simultaneous UAV having actively sweep angle morphing wing and flight control system design, *Aircr Eng. Aerosp. Technol*, 2023, **95**, pp 1062–1068.
[9] Hart, P.E., Nilsson, N.J. and Raphael, B. A formal basis for the heuristic determination of minimum cost paths, *IEEE Trans. Syst. Sci. Cybern.*, 1968, **4**, pp 100–107.
[10] Beamer, S., Asanović, K. and Patterson, D. Direction-optimizing breadth-first search, *Sci Program*, 2013, **21**, pp 137–148.
[11] LaValle, S. Rapidly-exploring random trees: a new tool for path planning, Research Report 9811, 1998.
[12] Kennedy, J. and Eberhart, R. Particle swarm optimization, *Proceedings of ICNN'95-International Conference on Neural Networks, November* 1995, 4, pp 1942–1948.
[13] Lambora, A., Gupta, K. and Chopra, K. Genetic algorithm – a literature review, *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, February2019, pp 380–384.
[14] Storn, R. and Price, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, *J Global Optim*, 1997, **11**, pp 341–359.
[15] Aslan, S., Demirci, S., Oktay, T. and Yesilbas, E. Percentile-based adaptive immune plasma algorithm and its application to engineering optimization, *Biomimetics*, 2023, **8**, p 486.
[16] Mirjalili, S., Mirjalili, S.M. and Lewis, A. Grey wolf optimizer, *Adv Eng Software*, 2014, **69**, pp 46–61.
[17] Dasdemir, E., Köksalan, M. and Öztürk, D.T. A flexible reference point-based multi-objective evolutionary algorithm: an application to the UAV route planning problem, *Comput. Oper. Res.*, 2020, **114**, p 104811.
[18] He, W., Qi, X. and Liu, L. A novel hybrid particle swarm optimization for multi-UAV cooperate path planning, *Appl. Intell.*, 2021, **51**, pp 7350–7364.
[19] Aslan, S. and Oktay, T. Path planning of an unmanned combat aerial vehicle with an extended-treatment-approach-based immune plasma algorithm, *Aerospace*, 2023, **10**, p 487.
[20] Shi, J., Tan, L., Zhang, H., Lian, X. and Xu,T. Adaptive multi-UAV path planning method based on improved gray wolf algorithm, *Comput Electr Eng*, 2022, **104**, p 108377.
[21] Huang, Z., Jiang, H., Wu, J., Liu, D., Dong, C. and Ping, X. Multi-UAV cooperative path planning via improved grey wolf optimization algorithm, *2024 IEEE International Conference on Unmanned Systems (ICUS)*, October 2024, pp 890–895.
[22] Lv, F. and Wang, L. Grey wolf optimizer for UAV path planning based on differential evolution mutation strategy, *2024 6th International Conference on Electronic Engineering and Informatics (EEI)*, June 2024, pp 72–77.
[23] Thoma, A., Thomessen, K., Gardi, A., Fisher, A and Braun, C. Prioritising paths: an improved cost function for local path planning for UAV in medical applications, *Aeronaut J*, 2023, **127**, pp 2125–2142.
[24] Cheng, Z., Tang, Y.X. and Liu, Y.L. 3-D path planning for UAV based on chaos particle swarm optimization, *Appl. Mech. Mater.*, 2012, **232**, pp 625–630.
[25] Kohli, M. and Arora, S. Chaotic grey wolf optimization algorithm for constrained optimization problems, *J Comput Des Eng*, 2018, **5**, pp 458–472.
[26] Yang, L., Guo, J. and Liu, Y. Three-dimensional UAV cooperative path planning based on the MP-CGWO algorithm, *Int. J. Innov. Comput. Inf. Control*, 2020, **16**, pp 991–1006.
[27] Wu, G., Mallipeddi, R. and Suganthan, P.N. Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization, *National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report*, 2017.
[28] Mirjalili, S. and Lewis, A. The whale optimization algorithm, *Adv Eng Software*, 2016, **95**, pp 51–67.
[29] Zhao, W., Wang, L. and Zhang, Z. Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm, *Neural Comput. Appl.*, 2020, **32**, pp 9383–9425.
[30] Dan, Z. and Long, Q. Three-dimensional route planning of UAV based on ACO-PSO algorithm, *J. Jilin Inst. Chem. Technol. J.*, 2017, **34**, pp 93–98.
[31] Xu, C., Xu, M. and Yin, C. Optimized multi-UAV cooperative path planning under the complex confrontation environment, *Comput Commun*, 2020, **162**, pp 196–203.

Cite this article: Zhang S. and Wang L. (2026). Multi-UAV cooperative path planning based on chaotic grey wolf optimiser. *The Aeronautical Journal*, **130**, 741–761. https://doi.org/10.1017/aer.2025.10101

https://doi.org/10.1017/aer.2025.10101 Published online by Cambridge University Press