scientific reports

Check for updates

OPEN

An improve crested porcupine algorithm for UAV delivery path planning in challenging environments

Shenglin Liu[1], Zikai Jin[1]✉, Hanting Lin[1] & Huimin Lu[2]

With the rapid advancement of drone technology and the growing applications in the field of drone engineering, the demand for precise and efficient path planning in complex and dynamic environments has become increasingly important. Traditional algorithms struggle with complex terrain, obstacles, and weather changes, often falling into local optima. This study introduces an Improved Crown Porcupine Optimizer (ICPO) for drone path planning, which enables drones to better avoid obstacles, optimize flight paths, and reduce energy consumption. Inspired by porcupines' defense mechanisms, a visuo-auditory synergy perspective is adopted, improving early convergence by balancing visual and auditory defenses. The study also employs a good point set population initialization strategy to enhance diversity and eliminates the traditional population reduction mechanism. To avoid local optima in later stages, a novel periodic retreat strategy inspired by porcupines' precise defenses is introduced for better position updates. Analysis on the IEEE CEC2022 test set shows that ICPO almost reaches the optimal value, demonstrating robustness and stability. In complex mountainous terrain, ICPO achieved optimal values of 778.1775 and 954.0118; in urban terrain, 366.2789 and 910.1682 and ranked first among the compared algorithms, proving its effectiveness and reliability in drone delivery path planning. Looking ahead, the ICPO will provide greater efficiency and safety for drone path planning in navigating complex environments.

Keywords Complex environments, Improved crested porcupine optimize, Path planning, UAV technology

With the rapid advancement of drone technology, the application of drones in logistics transportation has become increasingly widespread. Particularly in the realms of urban logistics, emergency rescue, and military transport, drones have emerged as critical tools for addressing complex transportation challenges due to their efficiency, flexibility, and convenience. However, planning transportation routes for drones in complex environments presents a formidable challenge. Such environments not only encompass geographical obstacles like skyscrapers, mountainous terrains, and rivers but also involve dynamic factors such as weather changes and traffic conditions. Consequently, devising efficient and safe transportation routes for drones in these complex settings has become an urgent problem that demands resolution.

The transportation path planning of unmanned aerial vehicles (UAVs) plays an increasingly vital role in contemporary logistics and transportation systems. As drone technology swiftly evolves and its potential in parcel and passenger transportation gains recognition, researchers and policymakers are fervently exploring methods to optimize drone transportation routes to bolster efficiency and reliability. Raghunatha et al. have proposed a policy development framework tailored for regional drone transportation systems, underscoring the significance of drones in urban and regional logistics. They highlighted the necessity of sound policies and frameworks, which are instrumental in facilitating the integration and utilization of drone technology and in navigating the intricacies and varied demands of drone operations within transportation systems[1]. Through a comprehensive literature review, Kellermann et al. analyzed the deployment of drones in parcel and passenger transport, pinpointing the pivotal role of path planning in enhancing transportation efficiency and curtailing operating costs. Particularly in urban settings, optimized path planning can dramatically reduce delivery times and alleviate traffic congestion[2]. Niu et al. investigated the resilient development of drone logistics from the

[1]College of Science and Technology Ningbo University, Ningbo, Zhejiang, China. [2]College of Architecture and Transportation Engineering, Ningbo University of Technology, Ningbo, Zhejiang, China. ✉email: jinzikai@nbu.edu.cn

perspectives of consumer choice, market competition, and regulatory impacts. Their research demonstrates that optimizing drone transportation paths in the face of uncertain market conditions and stringent regulatory frameworks can significantly fortify the resilience and responsiveness of logistics systems, ensuring their efficient functioning under diverse conditions[3]. Sun et al.'s research focuses on the deployment of coordinated scheduling between drones and riders within on-demand delivery services, proposing an innovative scheduling model that seeks to elevate delivery efficiency. This study accentuates the crucial nature of precise path planning in harmonizing the operations of drones with other transportation modes, especially within the complex urban landscapes where such coordination can substantially enhance the overall efficiency of logistics systems[4]. Frederiksen et al. projected the future applications of drones and their impact on policy-making up to the year 2057, offering a forward-looking perspective on the evolutionary trajectory of drone technology. They contend that future UAV transportation systems will require highly optimized path planning to adapt to the ever-evolving societal demands and technological progress[5]. Overall, UAV transportation path planning is increasingly vital in modern logistics, where optimizing routes enhances efficiency and reliability.

However, despite the valuable insights and substantial policy support furnished by existing research for drone transportation applications, path planning for UAVs continues to encounter significant challenges, particularly in navigating the complexities of highly dynamic and perpetually evolving urban or mountainous terrains. This predicament underscores the imperative for continued research into advanced path planning algorithms that can more adeptly tackle these challenges, thereby ensuring the optimality and practicality of drone transportation. Firstly, Ait Saadi et al. comprehensively reviewed various optimization approaches in UAV path planning, categorizing these methods into five major types: classical methods, heuristic methods, metaheuristic methods, machine learning, and hybrid algorithms. The paper systematically summarizes the recent research progress, technical challenges, and future research directions in this field, providing researchers with a comprehensive knowledge framework and technical guide[6]. Boulares et al. applied deep Q-learning to optimize UAV paths in marine environments, significantly improving search and rescue mission efficiency through autonomous adjustment. This method, while highly effective, demands substantial training data[7]. Meanwhile, Lee et al. employed goal-conditioned reinforcement learning, which uses subgoals for more granular path control, making it ideal for rapidly changing environments, though it may require complex goal-setting mechanisms[8]. Zhu et al. focused on multi-UAV operations with a deep Q-network approach to enhance energy efficiency and extend operational durations in IoT-assisted environments. Their method is highly efficient but depends on robust network conditions[9]. Lee et al. utilized a soft actor-critic model with hindsight experience replay, enhancing path planning and collision avoidance capabilities, albeit with complex model training requirements[10]. Swain et al. proposed a reinforcement learning-based cluster routing scheme for coordinated multi-UAV networks, improving network efficiency and responsiveness, yet necessitating effective cluster management mechanisms[11]. In addition to the path planning algorithm on the backend, there is also a path planning algorithm on the frontend aimed at dynamically optimizing the UAV movement trajectory in real-time. Huang et al. introduced an enhanced version of the Rapidly-exploring Random Tree (RRT) algorithm named Density Gradient-RRT. This improvement better handles spatial constraints and environmental variability, facilitating more efficient navigation in cluttered or unpredictable spaces. However, the algorithm might struggle with real-time adaptability in extremely dynamic environments[12]. Guo et al. proposed the Feedback RRT* algorithm for UAV path planning in hostile environments, integrating feedback mechanisms to adjust trajectories based on dynamic obstacles and threats. This approach significantly enhances navigational capabilities under adverse conditions but requires complex real-time feedback systems[13]. Feng et al. developed the DBVS-APF-RRT* algorithm, which combines the Artificial Potential Field and the RRT* framework to rapidly generate and refine paths, ensuring both speed and precision. Its ultra-high-speed path generation is a major advantage, but it may face challenges in extremely dense environments[14]. Rao et al. explored the Artificial Potential Field-A* algorithm for dual UAVs engaged in cooperative suspension transport. This method is ideal for scenarios requiring synchronized UAV movements but can be complex to implement and maintain synchronization in real-time[15]. Liu et al. developed a fast formation obstacle avoidance algorithm for clustered UAVs based on the Artificial Potential Field. This algorithm ensures the integrity of UAV formations while dynamically avoiding obstacles, making it useful for maintaining specific arrangements. The study by Guo et al. successfully optimizes path planning and routing efficiency in UAV ad hoc networks through the use of reinforcement learning, making a significant contribution to improving the performance of UAV networks in complex tasks and dynamic environments[16]. However, when dealing with larger-scale UAV networks, the real-time performance and scalability of the algorithm still require further optimization to ensure efficient path planning in highly dynamic scenarios. Similarly, Liu et al. propose a lightweight trustworthy message exchange mechanism that significantly enhances communication security in UAV networks, effectively addressing challenges in intelligent transportation systems[17]. Nonetheless, in the context of complex path planning, further improving this mechanism's adaptability and robustness in dynamic environments and potential path interference remains an important direction for future research. However, it may require extensive tuning to handle various environmental changes effectively[18]. Nonetheless, the most mainstream path planning algorithms currently are heuristic algorithms. These algorithms can plan the optimal path in an extremely short time, facilitating UAVs in transporting and delivering items. Recent studies have advanced this field further by incorporating various optimization techniques to enhance efficiency and accuracy in complex environments. Chen et al. introduced a Modified Central Force Optimization (MCFO) algorithm, optimizing 3D UAV path planning and showcasing significant improvements in path stability and obstacle avoidance[19]. Zhang et al. proposed a novel phase angle-encoded fruit fly optimization algorithm with a mutation adaptation mechanism, offering enhanced search capabilities and faster convergence to the optimal path[20]. Qu et al. developed a reinforcement learning-based Grey Wolf Optimizer algorithm, integrating learning capabilities into the optimization process for dynamic adjustments based on environmental feedback[21]. Avcu et al. discussed a Whale Optimization Algorithm (WOA)-based method focused on preventing collisions in

cluttered environments, essential for real-world applications like disaster management[22]. Yu et al. tackled path planning in disaster scenarios using a constrained differential evolution algorithm, emphasizing robust solutions in emergency conditions[23]. Huang et al. introduced an Adaptive Cylinder Vector Particle Swarm Optimization combined with Differential Evolution, refining the UAV path planning strategy in unpredictable environments[24]. Chai et al. presented a Multi-strategy Fusion Differential Evolution algorithm, integrating multiple strategies to optimize path planning in complex scenarios, highlighting its adaptability[25]. Hu et al. improved upon the Honey Badger Algorithm by incorporating multiple strategies, resulting in the SaCHBA_PDN algorithm that enhances path planning efficiency[26]. Similarly, Zhang, Zhou, Qin, and Tang developed Heuristic Crossing Search and Rescue Optimization, tailored for search and rescue missions, demonstrating practical applications in emergency scenarios[27]. Lastly, these innovative approaches highlight the ongoing advancements in UAV path planning, demonstrating the field's dynamic evolution and its critical role in enhancing UAV operational capabilities. Although these innovative methods demonstrate the dynamic development in the field, the performance and efficiency of existing heuristic algorithms are still affected in complex and unpredictable environments, requiring significant improvements.

In addressing specialized challenges such as UAV path planning, existing heuristic algorithms continue to require enhancements to boost performance and efficiency. Recently, scholars have developed various improved heuristic algorithms aimed at enhancing the optimization capabilities of diverse systems and models. Karthik et al. proposed an enhanced crocodile optimization algorithm, refining hunting and mating strategies to optimize heterogeneous UAV coverage path planning, thereby improving coverage efficiency and energy savings during mission execution[28]. Qadir et al. studied collision-free path planning for UAVs in disaster situations and demonstrated the superior performance of the improved DGBCO algorithm in various complex environments through comparisons. Although the study considered multiple environmental scenarios, it lacked research on obstacle dynamics in complex environments, which somewhat limits the applicability of the method[29]. Fan et al. employed a cyclic chaotic map, levy flight operator, and non-linear adaptive weights to refine the tuna swarm optimization algorithm, successfully addressing the flexible job shop scheduling problem with random machine failures[30]. Abdel-Basset et al. used a novel Lévy-Normal mechanism to enhance the Kepler optimization algorithm for selecting optimal parameters of proton exchange membrane fuel cells, demonstrating favorable outcomes in comparative studies[31]. Pan et al. addressed optimal scheduling of orderly charging and discharging for electric vehicles by combining the particle swarm optimization algorithm with the gravitational search algorithm, significantly reducing users' charging costs and electrical grid load variations, improving grid stability and user economic benefits[32]. Pan et al. optimized a rolling bearing dynamic model and integrated sensitive features using a golden jackal optimization algorithm with dimension-by-dimension inverse learning strategy and adaptive weights[33]. Krishnan et al. improved the Archimedes optimization algorithm by incorporating an increased density decrement factor, enhancing performance and efficiency of solar cells[34]. SeyedGarmroudi et al. enhanced the pelican optimization algorithm by modifying three movement strategies and a predefined knowledge sharing factor to solve load scheduling problems, thereby increasing the algorithm's solution precision and efficiency[35]. Pan et al. developed an improved artificial bee colony algorithm based on a two-dimensional queue structure, effectively maintaining performance in scenarios with large data volumes[36]. These advancements represent significant strides in heuristic algorithms, reflecting ongoing efforts to refine techniques across various applications for improved optimization and efficiency. To summarize, the main differences between these improved methods lie in the diversity of optimization strategies, the innovation in algorithm structures, and the specificity of application scenarios. In terms of optimization strategies, some methods enhance the search capability and global optimization ability of the algorithms by introducing complex mathematical mechanisms such as chaotic mapping and Levy flight operators, while others combine strategies like reverse learning and tabu search to avoid local optima and improve optimization efficiency. Regarding algorithm structure, methods such as adjusting initialization strategies, introducing adaptive weights, or improving movement strategies are used to enhance the dynamic response and adaptability of the algorithms, optimizing the solution accuracy for different problems. Lastly, these methods also differ in their application scenarios, with each method specifically tailored to improve the algorithm to meet the demands of specific problems, thereby increasing efficiency and optimization effectiveness. However, despite these advancements, further improvements are still needed to address limitations such as computational complexity and the ability to handle increasingly complex and dynamic environments effectively.

In summary, this study introduces an improved Crested Porcupine Optimizer (ICPO) for UAV delivery path planning in complex environments. Unlike the existing methods that often rely on complex mathematical mechanisms such as chaotic mapping and Levy flight operators, inspired by the unique defensive behavior of the crested porcupine, this study posits that vision and hearing are mutually beneficial. By adopting an audiovisual reciprocal defense mechanism, it improves upon the original framework where the author considered vision as the primary defense mechanism and hearing as the secondary defense mechanism, addressing the key issue of slow early convergence in traditional CPO. Furthermore, inspired by the crested porcupine's ability to use its defense mechanisms equally at all positions, a good point set population initialization strategy is employed to increase population diversity and removed the population reduction mechanism based on the characteristics of the average distribution of the population. To address the issue of traditional CPO getting easily trapped in local optima during later stages, this study proposes a novel periodic retreat strategy to improve position updates. This strategy inspired by the crested porcupine's refined defensive mechanisms. Unlike other methods that rely on strategies like reverse learning and tabu search, this approach significantly enhances the algorithm's ability to escape local optima as it progresses. Ultimately, this optimizer can achieve higher efficiency and excellent path selection capabilities, effectively solving the challenges of UAV delivery path planning in complex environments. The UAV delivery path planning is shown in Fig. 1:

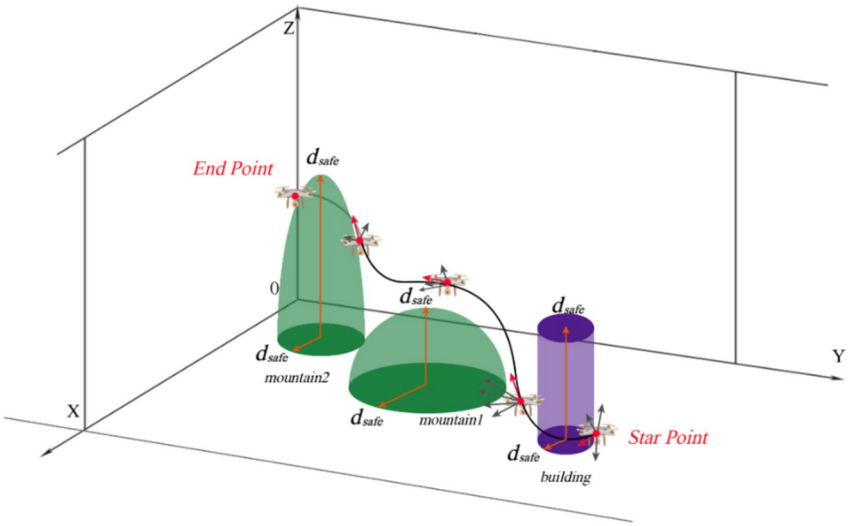

Fig. 1. Drone delivery path planning diagram.

The rest of this paper is organized as follows. "Model establishment" introduces the model structure for UAV path planning, the establishment of the CPO, and the scientific improvements and their purposes. "Experimental analysis" presents the efficiency of the proposed ICPO on the test set and the experimental results of UAV delivery paths in complex areas, comparing its performance with the latest advanced methods. "Conclusion" provides a summary of this study.

Model establishment

Drone path planning model

The drone path planning model aims to determine an optimized path for a drone to efficiently complete a mission while adhering to certain constraints. This section details the models for travel distance, altitude change, horizontal and vertical turning angle changes, and the associated constraints, forming the basis of the optimization problem.

Travel distance model

The travel distance model calculates the total distance the drone travels from the starting point to the destination. Let the starting point be $A(x_A, y_A, z_A)$ and the endpoint be $B(x_B, y_B, z_B)$. For any point $P_i(x_i, y_i, z_i)$ on the path, the travel distance D is the sum of the distances between consecutive points:

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

where, n is the number of points contained in the path that is usually determined the number of algorithm iterations and $n - 1$ refers to the number of path segments calculated after removing the starting point.

The behavior of drone delivery path length is shown in the Fig. 2:

Altitude change model

The altitude change model ensures the drone follows a predefined altitude profile to adapt to terrain and other flight requirements. Given an altitude function $z(x, y)$ representing the path's altitude change, the altitude changes Δz for points $P_i(x_i, y_i, z_i)$ on the path is:

$$\Delta z = \sum_{i=1}^{n-1} |z_{i+1} - z_i| \tag{2}$$

where, n has the same meaning as in Eq. (1).

The height behavior of the drone delivery path is shown in the Fig. 3:

Turning angle change model

The turning angle change model ensures smooth turns during the flight, avoiding sharp turns that could compromise safety and increase energy consumption. This includes both horizontal and vertical turning angles. For points P_{i-1}, P_i, P_{i+1} on the path:

Horizontal turning angle θ_i:

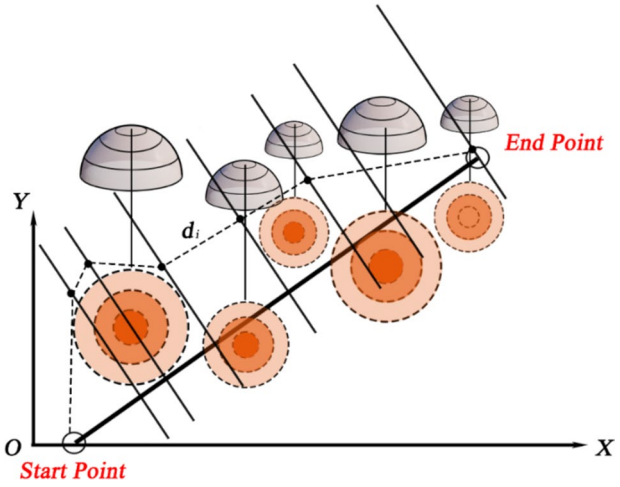

Fig. 2. Behavior diagram of drone delivery path length.

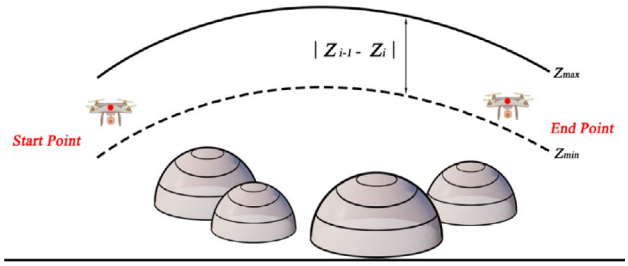

Fig. 3. Drone delivery path height behavior diagram.

$$\cos(\theta_i) = \frac{(x_i - x_{i-1})(x_{i+1} - x_i) + (y_i - y_{i-1})(y_{i+1} - y_i)}{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \tag{3}$$

$$\Delta\theta = \sum_{i=2}^{n-1} |\theta_i - \theta_{i-1}| \tag{4}$$

Vertical turning angle ϕ_i:

$$\cos(\phi_i) = \frac{(z_i - z_{i-1})(z_{i+1} - z_i)}{\sqrt{(z_i - z_{i-1})^2}\sqrt{(z_{i+1} - z_i)^2}} \tag{5}$$

$$\Delta\phi = \sum_{i=2}^{n-1} |\phi_i - \phi_{i-1}| \tag{6}$$

The turning behavior of the drone delivery path is shown in the Fig. 4:

Objective function
The objective function integrates travel distance, altitude change, and turning angle changes to form a multi-objective optimization problem. The function to be minimized is:

$$\min (D + \Delta z + \Delta\theta + \Delta\phi) \tag{7}$$

Constraints
The path planning must satisfy several constraints to ensure safety and feasibility:

- Safety Constraints: Ensure the distance d_i between any path point P_i and obstacle $O_j(x_j, y_j, z_j)$ is greater than the safety distance d_{safe}:

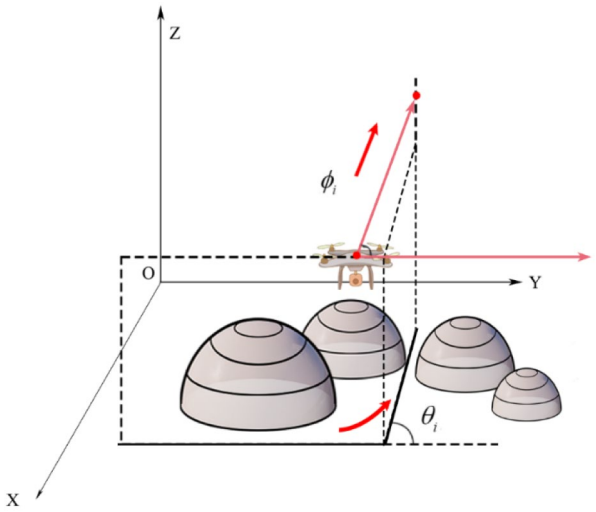

Fig. 4. Drone delivery path turning behavior diagram.

$$d_i = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{8}$$

Using the penalty function method, the penalty term P_{safe} is introduced:

$$P_{safe} = \begin{cases} 0, & if \quad d_i > d_{safe} \\ \infty, & if \quad d_i < d_{safe} \\ d_{safe} - d_i, & otherwise \end{cases} \tag{9}$$

- Path Smoothness Constraints: Use cubic spline interpolation to ensure path smoothness. For path points $\{P_0, P_1, \ldots, P_n\}$, the cubic spline function $S_i(x)$ between consecutive points is:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \tag{10}$$

The conditions are:

$$S_i(x_i) = P_i \tag{11}$$

$$S_i(x_{i+1}) = P_{i+1} \tag{12}$$

$$S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}) \tag{13}$$

$$S_i''(x_{i+1}) = S_{i+1}''(x_{i+1}) \tag{14}$$

The schematic diagram after cubic spline interpolation is as follows Fig. 5:

- Altitude Constraints: Ensure altitude changes are within acceptable ranges and above the terrain, yet below the drone's maximum altitude z_{max}:

$$z_{terrain}(x_i, y_i) \le z_i \le z_{max} \tag{15}$$

$$\Delta z \le \Delta z_{\max} \tag{16}$$

Using the penalty function method, the penalty term P_{alt} is introduced:

$$P_{alt} = \begin{cases} 0, & if \quad z_i > z_{max} \\ \infty, & if \quad z_i < z_{terrain}(x_i, y_i) \\ z_{max} - z_{terrain}(x_i, y_i), & otherwise \end{cases} \tag{17}$$

- Turning Angle Constraints: Ensure horizontal and vertical turning angles are within acceptable ranges:

$$0 < \theta_i \le \theta_{\max} \tag{18}$$

$$0 < \phi_i \le \phi_{\max} \tag{19}$$

Using the penalty function method, the penalty term P_{angle} is introduced:

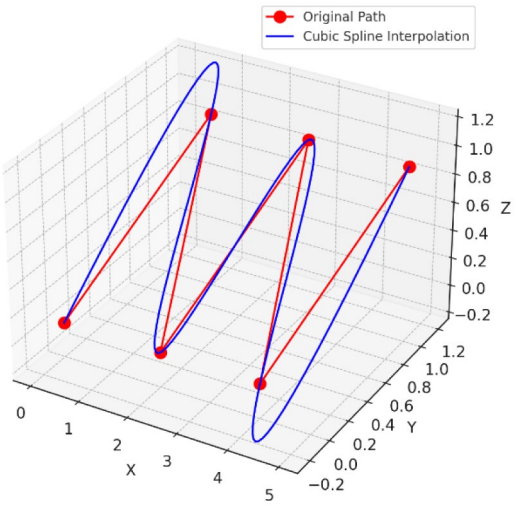

Fig. 5.. 3D View of Original Path and Cubic Spline Interpolation.

$$P_{angle} = \begin{cases} 0, & if \ \ \theta_i > \theta_{\max} \ \ and \ \ \phi_i > \phi_{\max} \\ \infty, & if \ \ \theta_i \leq 0 \ \ and \ \ \phi_i \leq 0 \\ \theta_i - \theta_{\max} + \phi_i - \phi_{\max}, & otherwise \end{cases} \tag{20}$$

Integrating the above models and constraints, the optimization problem is formulated as follows:

$$\min \left(D + \Delta z + \Delta\theta + \Delta\phi + P_{safe} + P_{alt} + P_{angle}\right)$$
$$s.t. \begin{cases} S_i(x_i) = P_i \\ S_i(x_{i+1}) = P_{i+1} \\ S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}) \\ S_i''(x_{i+1}) = S_{i+1}''(x_{i+1}) \\ z_{terrain}(x_i, y_i) \leq z_i \leq z_{\max} \\ \Delta z \leq \Delta z_{\max} \\ 0 < \theta_i \leq \theta_{\max} \\ 0 < \phi_i \leq \phi_{\max} \\ \Delta\theta \leq \Delta\theta_{\max} \\ \Delta\phi \leq \Delta\phi_{\max} \end{cases} \tag{21}$$

Crested Porcupine Optimizer

The Crested Porcupine Optimizer (CPO) is a novel metaheuristic algorithm inspired by the defensive behaviors of the crested porcupine[37]. Unlike other algorithms that often mimic offensive behaviors, CPO leverages the crested porcupine's four distinct defensive strategies—sight, sound, odor, and physical attack—to balance exploration and exploitation effectively in the search space.

The optimization process begins with the initialization of a population $\vec{X}_i$ comprising N' candidate solutions. Each candidate solution is initialized within the boundaries of the search space:

$$\vec{X}_i = \vec{L} + \vec{r} \times \left(\vec{U} - \vec{L}\right), \quad i = 1, 2, \ldots, N' \tag{22}$$

where $\vec{L}$ and $\vec{U}$ represent the lower and upper bounds of the search space, respectively, and $\vec{r}$ is a random vector uniformly distributed between 0 and 1.

To enhance population diversity and accelerate convergence, CPO employs a cyclic population reduction technique. This technique adjusts the population size N dynamically during the optimization process:

$$N = N_{\min} + (N' - N_{\min}) \times \left(1 - \left(\frac{t\% \frac{T_{\max}}{T}}{\frac{T_{\max}}{T}}\right)\right) \tag{23}$$

where T denotes the cyclic variable determining the number of cycles, t is the current function evaluation count, T_{max} is the maximum number of function evaluations, and N_{min} is the minimum allowable population size. This approach simulates the idea that not all porcupines activate their defense mechanisms simultaneously, thus preserving diversity and enhancing convergence.

The exploration phase of CPO is modeled after the sight and sound defense mechanisms of the crested porcupine, aimed at surveying different regions of the search space.

- First Defense Mechanism (Sight): When a crested porcupine detects a predator from a distance, it raises and fans its quills to appear larger. Mathematically, this behavior is represented as:

$$\overrightarrow{x_i^{t+1}} = \overrightarrow{x_i^t} + \tau_1 \times \left| 2 \times \tau_2 \times \overrightarrow{x_{CP}^t} - \overrightarrow{y_i^t} \right| \tag{24}$$

$$\overrightarrow{y_i^t} = \frac{\overrightarrow{x_i^t} + \overrightarrow{x_r^t}}{2} \tag{25}$$

where, $\overrightarrow{x_{CP}^t}$ represents the best obtained solution. The vector $\overrightarrow{y_i^t}$ is generated midway between the current best solution and a randomly selected solution from the population, representing the position of a predator at iteration t. The parameter τ_1 is a random number based on the normal distribution, and τ_2 is a random value in the interval [0,1]. After that, r represents a randomly selected number from the range [1,N] which is why $\overrightarrow{x_r^t}$ corresponds to the position of the solution within [1,N].

- Second Defense Mechanism (Sound): If the sight mechanism does not deter the predator, the porcupine makes various sounds. This mechanism is modeled as:

$$\overrightarrow{x_i^{t+1}} = \left(1 - \overrightarrow{U_1}\right) \times \overrightarrow{x_i^t} + \overrightarrow{U_1} \times \left(\vec{y} + \tau_3 \times \left(\overrightarrow{x_{r1}^t} - \overrightarrow{x_{r2}^t}\right)\right) \tag{26}$$

where, r_1, r_2 are two random integers chosen from the range [1,N], and τ_3 is a random value generated within the interval [0,1].

When the predator persists, the porcupine resorts to more aggressive defense mechanisms, which correspond to the exploitation phase.

- Third Defense Mechanism (Odor): If the first two mechanisms fail, the porcupine emits a foul odor to repel the predator. This behavior is mathematically expressed as:

$$\overrightarrow{x_i^{t+1}} = \left(1 - \overrightarrow{U_1}\right) \times \overrightarrow{x_i^t} + \overrightarrow{U_1} \times \left(\overrightarrow{x_{r_1}^t} + S_i^t \times \left(\overrightarrow{x_{r_2}^t} - \overrightarrow{x_{r_3}^t}\right) - \tau_3 \times \vec{\delta} \times \gamma_t \times S_i^t\right) \tag{27}$$

$$\vec{\delta} = \begin{cases} 1, & \overrightarrow{rand} \leq 0.5 \\ -1, & Else \end{cases} \tag{28}$$

$$\gamma_t = 2 \times rand \times \left(1 - \frac{t}{t_{max}}\right)^{\frac{t}{t_{max}}} \tag{29}$$

$$S_t^i = \exp\left(\frac{f\left(x_t^i\right)}{\sum_{k=1}^{N} f\left(x_t^k\right) + \varepsilon}\right) \tag{30}$$

where, r_3 is a random number between [1,N], and $\vec{\delta}$ is a parameter that controls the search direction, $\overrightarrow{x_i^t}$ represents the position of the *i-th* individual at iteration t. γ_t is the defense factor, τ_3 is a random value within the interval [0,1], and S_t^i is the odor diffusion factor.

- Fourth Defense Mechanism (Physical Attack): As a last resort, the porcupine physically attacks the predator. This aggressive strategy is represented as:

$$\overrightarrow{x_i^{t+1}} = \overrightarrow{x_{CP}^t} + (\alpha(1 - \tau_4) + \tau_4) \times \left(\delta \times \overrightarrow{x_{CP}^t} - \overrightarrow{x_i^t}\right) - \tau_5 \times \delta \times \gamma_t \times \overrightarrow{F_i^t} \tag{31}$$

$$\overrightarrow{F_i^t} = \overrightarrow{\tau_6} \times \frac{m_i \times \left(\overrightarrow{v_i^{t+1}} - \overrightarrow{v_i^t}\right)}{\Delta t} \tag{32}$$

$$m_i = e^{\frac{f\left(\overrightarrow{x_i^t}\right)}{\sum_{k=1}^{N} f\left(\overrightarrow{x_k^t}\right) + \varepsilon}} \tag{33}$$

$$\overrightarrow{v_i^{t+1}} = \overrightarrow{x_r^t} \tag{34}$$

$$\overrightarrow{v_i^t} = \overrightarrow{x_i^t} \tag{35}$$

where, $\overrightarrow{x_{CP}^t}$ is the best-obtained solution, representing the central point (CP). The position $\overrightarrow{x_i^t}$ refers to the location of the *i-th* individual at iteration *t*, representing the predator's position. The parameter α is a convergence speed factor, which will be discussed later in the parameter settings section. τ_4 is a random value within the interval [0,1]. The vector $\overrightarrow{F_i^t}$ represents the average force exerted by the CP on the *i-th* predator, m_i is the mass of *i-th* individual (predator) at iteration *t*, $\overrightarrow{v_i^{t+1}}$ represents the final speed of the *i-th* individual at the next iteration $t+1$ and is determined by selecting a random solution from the current population. The variable $\overrightarrow{v_i^t}$ denotes the initial speed of the *i-th* individual at iteration *t*. The term Δt represents the number of the current iteration. The vector $\overrightarrow{\tau_6}$ consists of random values generated within the interval [0,1].

Each candidate solution's fitness is evaluated using the objective function. The best solution X_{best} is updated iteratively based on the new fitness values. The optimization process continues until the termination criteria, such as a maximum number of iterations or a satisfactory fitness level, are met.

Improvement strategy

To address the issues of slow convergence, inadequate optimization performance, and unreasonable resource allocation in the CPO, this study proposes a series of improvement measures.

Population initialization

Population initialization is a crucial step in the performance of the Crested Porcupine Optimizer (CPO). A well-initialized population can significantly enhance the convergence speed and the exploration–exploitation balance. Inspired by the crested porcupine adopts a good point set population initialization method to improve the diversity and quality of the initial solutions[38]. Just as the crested porcupine uses a variety of defenses to handle different threats effectively, this initialization method ensures a robust and varied starting population, leading to more efficient and effective optimization.

The concept of a Good Point Set involves creating a sequence of points within a unit cube in *D*-dimensional Euclidean space. These points, known as a "Good Point Set" are chosen for their even distribution and low discrepancy. To generate a Good Point Set for any integer G_D, a specific sequence is used. This method involves a simple formula: the coordinates of each point are obtained by multiplying a factor *r* by the index *k* and then taking the fractional part (denoted as $\{\cdot\}$). Here, *k* is an integer ranging from 1 to *n*, where *n* represents the total number of points generated.

A key feature of the Good Point Set is its very low discrepancy, meaning that its distribution is very close to perfectly uniform. The exact size of the discrepancy is given by the formula $\varphi(n) = C(r,\varepsilon)n^{-1+\varepsilon}$.In this formula, ε is a very small positive number, and $C(r,\varepsilon)$ is a constant that depends on *r* and ε.This indicates that as the number of points *n* increases, the discrepancy decreases.

To generate these special Good Points, this study opted for a method based on prime numbers p and the cosine function to determine the value of r, with the calculation method as follows:

$$r = 2\mathrm{Cos}\left(\frac{2k\pi}{p}\right),\ k \in [1,\mathrm{D}] \tag{36}$$

$$p = \min\{x \in P : \frac{x-3}{2} \geq D\} \tag{37}$$

where, *p* is a specific prime number, the smallest prime number that satisfies condition $\frac{p-3}{2} \geq D$.

This selection guarantees that the Good Points are not just evenly spread out but also have favorable mathematical qualities, making them efficient in covering the entire search space in multiple dimensions. This is demonstrated in Fig. 6.

Removing population reduction

In this step, the algorithm eliminates the process of population reduction to maintain a consistent number of points throughout its execution. Similar to how the crested porcupine maintains a series of continuous and consistent defensive behaviors when facing threats, this approach ensures a stable and uniform exploration of the search space by keeping the population size constant instead of reducing the number of points as iterations progress. This strategy allows for continuous sampling of all regions, thereby preventing the loss of potential solutions. The method is as follows:

$$N(t) = N \tag{38}$$

Mutually beneficial audiovisual defense mechanism

Inspired by the strategic use of vision and sound by crested porcupines to deter threats, crested porcupines should employ a combined visual and auditory defense. The Symbiotic Organisms Search (SOS) algorithm is an algorithm inspired by the symbiotic phenomena in biology[39]. By integrating the mutually beneficial position update logic of SOS with the visual and auditory defenses of crested porcupines (CPO), a new solution is created that balances exploration and rapid convergence. This approach ensures that CPOs maintain a wide search in the early stages to avoid local optima and accelerate convergence. The specific formulas are as follows:

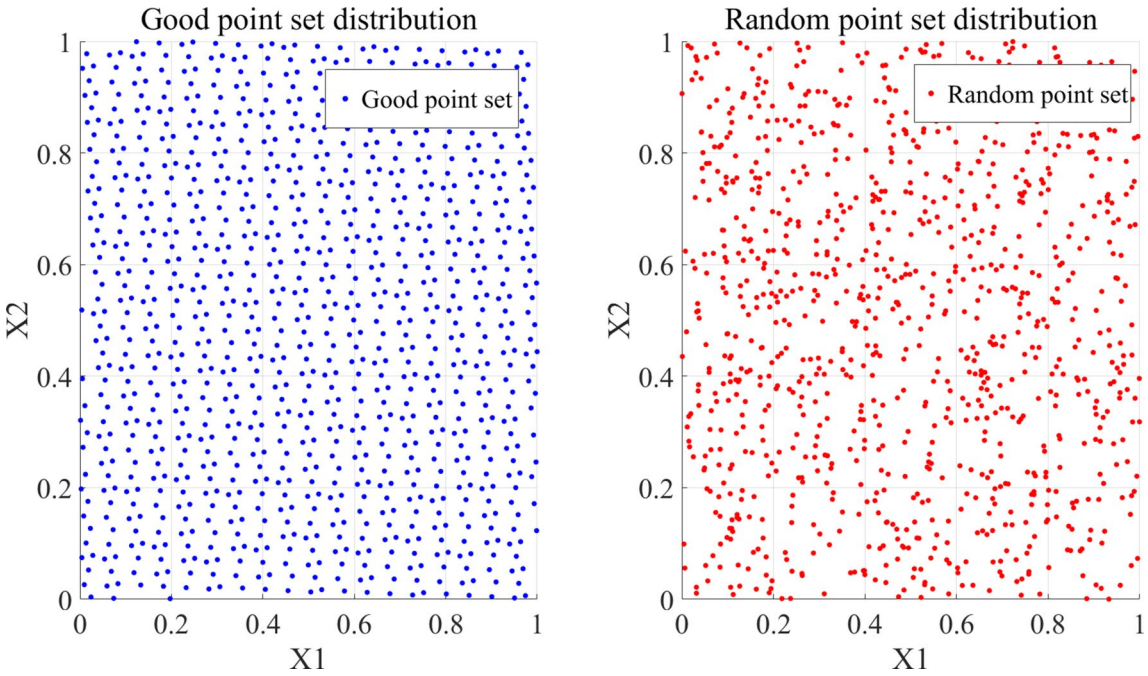

Fig. 6. Comparison of population initialization effects.

- Mutually beneficial stage:

$$\overrightarrow{R_{MV}} = \overrightarrow{x_i^t} + rand \cdot \left(\overrightarrow{x_{rand}} - \overrightarrow{x_i^t}\right) \tag{39}$$

$$\overrightarrow{x_i^{t+1}} = \overrightarrow{x_i^t} + rand \cdot \left(\overrightarrow{x_{CP}^t} - \overrightarrow{R_{MV}}\right) \tag{40}$$

where, $\overrightarrow{x_{rand}}$ is the randomly selected partner individual, and $\overrightarrow{R_{MV}}$ is the mutualism parameter.

- Joint defense stage:

$$\overrightarrow{R_{CM}} = \overrightarrow{x_i^t} + rand \cdot \left(\overrightarrow{x_{CP}^t} - \overrightarrow{x_{rand}}\right) \tag{41}$$

where, $\overrightarrow{R_{CM}}$ is the joint defense parameter and the position update formula is as follows:

$$\begin{cases} \overrightarrow{x_i^{t+1}} = \overrightarrow{R_{CM}}, & f_{\left(\overrightarrow{x_i^t}\right)} < f_{\left(\overrightarrow{x_{rand}}\right)} \\ \overrightarrow{x_i^{t+1}} = \overrightarrow{x_i^t}, & else \end{cases} \tag{42}$$

Periodic retreat

This study proposes a periodic retreat strategy to simulate the defensive tactics employed by crested porcupines when facing threats. Inspired by the unique defense mechanisms of crested porcupines, this strategy addresses the issue of CPO getting trapped in local optima in later stages. By implementing a periodic retreat strategy with non-linear weight controlling the retreat step length, the CPO are helped to escape local optima, thereby enhancing the algorithm's exploitation capabilities. The specific formulas are as follows:

Random radius calculation:

$$r_g = 0.9 - \log(\mathrm{t}+1) \cdot \frac{(0.9-0.1)}{\log(t_{\max}+1)} \tag{43}$$

$$r = rand \cdot r_g \tag{44}$$

where, $t_{\max}$ is the maximum number of time steps.

- Random differential vector calculation:

$$L = \overrightarrow{x_{CP}^{t}} - \overrightarrow{x_{i}^{t}} \tag{45}$$

$$L_p = L \cdot rand(1, \dim) \tag{46}$$

- Evacuation step size calculation

$$\alpha = L^2 + L_p^2 - 2LL_p \cos\left[2\pi \cdot rand(1, \dim)\right] \tag{47}$$

$$\overrightarrow{x_{i}^{t+1}} = \overrightarrow{x_{i}^{t}} + r \cdot \alpha, if \ \mathrm{mod}\left[t, round\left(\frac{t_{\max}}{10}\right)\right] == 0 \tag{48}$$

Through the above improvement strategies, CPO can efficiently converge in a short period of time. The specific flowchart is as follows Fig. 7.

Experimental analysis

Environmental configuration

In this experiment, this study will use ICPO and compare it with other advanced algorithms for UAV delivery path planning in complex environments. The focus of the study is to ensure the consistency of initial parameter settings for the algorithms to guarantee reproducibility. The selection of these parameters is based on a comprehensive understanding of the problem domain and insights from previous research, aiming to enhance the effectiveness and generalization capability of the algorithms across various datasets and scenarios.

The hardware configuration used for the simulation is detailed in Table 1.

To ensure the reproducibility of the experiment, it is crucial to consistently set the initial parameters of the algorithm. Parameter selection is informed by domain knowledge and previous research to guarantee the algorithm's effectiveness and generalizability across various datasets and scenarios. The initial parameter settings for several key algorithms are outlined in Table 2.

Test set testing

In this study, the performance of ICPO was evaluated using Table 3 [40], which includes 12 international standard test functions from IEEE CEC2022. These test functions include unimodal functions, multimodal functions, and composite benchmark functions, which are used to assess the algorithm's local search ability, global search ability, and exploration and exploitation capabilities, respectively. In addition to comparing with the traditional CPO, the study also compared ICPO with the classic Whale Optimization Algorithm[41] (WOA), the top-performing Parrot Optimizer[42] (PO), the Black-winged Kite Algorithm[43] (BKA) and the coati optimization algorithm[44] (COA) providing a comprehensive understanding of ICPO's performance relative to industry standards.

This study selected the IEEE CEC2022 test set with a dimension of 20, a population size of 50, and 200 iterations as the unified testing indicator for evaluating effectiveness. The convergence curves of the ICPO and other comparable algorithms for various functions are shown in Fig. 8.

Repeat the above algorithm 50 times to obtain the result:

The results from Table 4 indicate that the ICPO algorithm consistently outperforms other comparison algorithms in terms of mean and standard deviation values across most test functions. Specifically, ICPO achieved the lowest mean values for all functions except F4, demonstrating its superior optimization performance and stability. Notably, for function F6, the ICPO achieved a mean value of 6.76E+03, which is significantly lower than that of other algorithms. Moreover, ICPO achieved the lowest standard deviation values for functions F1, F2, F3, F6, F7, F8, F9, and F10, indicating its high robustness and consistency. These findings highlight the effectiveness of the improved ICPO algorithm in solving complex optimization problems.

Following this promising performance, the next step involves applying the ICPO algorithm to UAV delivery path planning. This application will test the algorithm's practical utility in optimizing real-world logistics and navigation tasks.

Terrain generation model

- Mountainous Terrain Generation:

The formula for generating mountainous terrain is given by:

$$\begin{aligned} z = {} & \sin(y + a) + b\sin(x) + c\cos(d(y^2 + x^2)) \\ & + e\cos(y) + f\sin(f(y^2 + x^2)) + g\cos(y) \end{aligned} \tag{49}$$

where $a = 1, b = 1, c = 1, d = 1, e = 1, f = 1$, and $g = 1$ are terrain constants. The base height $h = 0$ is set to 0, representing the lowest position of the mountain peaks. The overall height h is calculated as:

$$h = \sum_{i} \left(h_i \times \exp\left(-\frac{(x - A_{i,1})^2}{a_i^2} - \frac{(y - A_{i,2})^2}{b_i^2} \right) \right) + h_0 \tag{50}$$

Finally, the value of the terrain is given by:

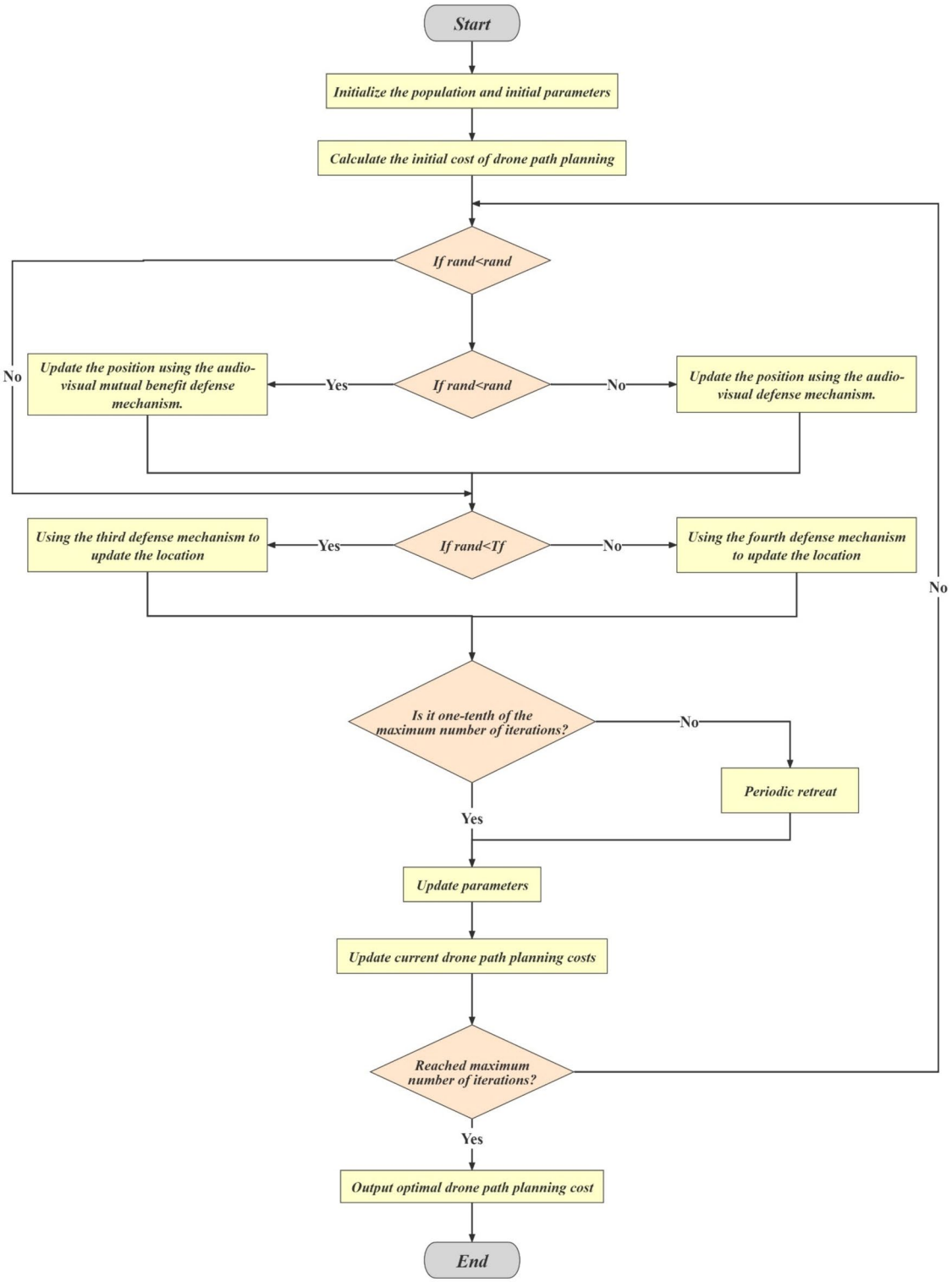

Fig. 7. ICPO algorithm flowchart.

$$\text{Value} = \max(z, h) \tag{51}$$

- Urban Terrain Generation
  The formula for generating urban terrain is given by:
  To calculate the height of the buildings:

Configuration	Description
CPU Model	Intel i7-11800H
Operating System	Windows 10 Pro
System Memory	16.0 GB
GPU Model	NVIDIA RTX-3060
GPU Memory	13.9 GB
Code Environment	MATLAB 2023b

Table 1. Environmental configuration.

Algorithm Name	Parameter	Initial Setting
CPO	Update cycle	2
	Population reduction	0.8
	The proportion of first and second defense mechanisms	Random
	The proportion of third and fourth defense mechanisms	0.8
ICPO	Update cycle	2
	Population reduction	None
	The proportion of first and second defense mechanisms	Random
	The proportion of third and fourth defense mechanisms	0.8

Table 2. Algorithm parameters.

	No.	Functions	F_i^*
Unimodal function	1	Shifted and full Rotated Zakharov Function	300
Basic functions	2	Shifted and full Rotated Rosenbrock's Function	400
	3	Shifted and full Rotated Expanded Schaffer's f_6 Function	600
	4	Shifted and full Rotated Non-Continuous Rastrigin's Function	800
	5	Shifted and full Rotated Levy Function	900
Hybrid functions	6	Hybrid Function 1 ($N = 3$)	1800
	7	Hybrid Function 2 ($N = 6$)	2000
	8	Hybrid Function 3 ($N = 5$)	2200
Composition functions	9	Composition Function 1 ($N = 5$)	2300
	10	Composition Function 2 ($N = 4$)	2400
	11	Composition Function 3 ($N = 5$)	2600
	12	Composition Function 4 ($N = 6$)	2700
Search range: $[-100, 100]^D$			

Table 3. Specific situation of CEC2022 test set.

$$h = \max\left((\sqrt{(x - bx_i)^2 + (y - by_i)^2} \leq br_i) \times bh_i \right) \tag{52}$$

where bx_i and by_i are the coordinates of the *i-th* building's location, br_i is the radius of the *i-th* building, and bh_i is the height of the *i-th* building. The term inside the max function checks if the point (x, y) is within the radius of the building, and if so, returns the building's height bh_i.

The final map value is then given by:

$$\text{Value} = h \tag{53}$$

These formulas allow for the generation of both mountainous and urban terrains, essential for simulating diverse environments in UAV delivery path planning.

Comparative experiment

To closely simulate real-world environments and perform sensitivity analysis on algorithm performance, this study constructs three types of six 3D maps, utilizing the parameters given in Table 5. For each scenario, multiple obstacles are incorporated to increase the environmental complexity.

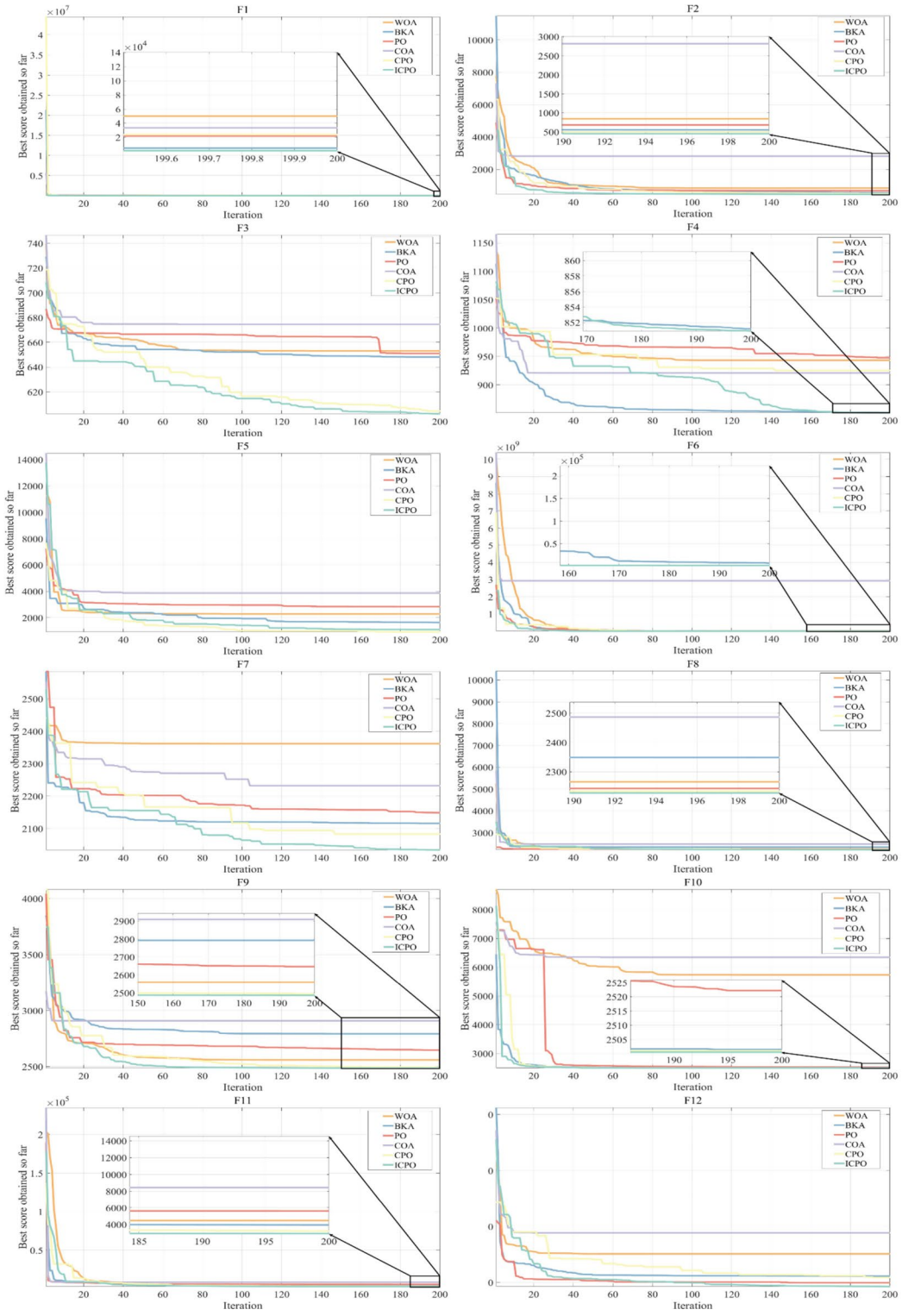

Fig. 8. Comparison of algorithm convergence curves.

Tables 6, and 7 summarizes the details of each case. To validate algorithm performance, this study compares the fitness values and the obtained path lengths of each algorithm across all scenarios, reporting the performance and results for each case.

To conduct a simulation comparison of UAV delivery path planning using recent advanced algorithms: A hybrid algorithm based on modified mayfly algorithm[45] (MODMA), grey wolf optimizer and differential evolution[46] (HGWODE), a novel hybrid chaotic aquila optimization algorithm with simulated annealing[47] (CAOSA) and Symbiotic organisms search and sine–cosine particle swarm optimization[48] (HISOS-SCPSO).

	WOA		BKA		PO		COA		CPO		ICPO	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	4.37E+04	1.62E+04	7.14E+03	3.48E+03	1.66E+04	5.07E+03	4.45E+04	1.26E+04	2.39E+04	4.31E+03	**1.39E+03**	**7.99E+02**
F2	6.88E+02	9.12E+01	6.27E+02	1.42E+02	6.86E+02	1.05E+02	2.39E+03	5.86E+02	4.89E+02	1.38E+01	**4.59E+02**	**1.12E+01**
F3	6.66E+02	1.17E+01	6.52E+02	9.57E+00	6.54E+02	1.01E+01	6.77E+02	8.83E+00	6.08E+02	1.57E+00	**6.02E+02**	**1.45E+00**
F4	9.47E+02	3.01E+01	8.84E+02	1.99E+01	9.14E+02	2.33E+01	9.69E+02	1.68E+01	9.26E+02	**9.42E+00**	**8.55E+02**	1.80E+01
F5	4.54E+03	1.48E+03	2.04E+03	3.56E+02	2.68E+03	4.11E+02	3.39E+03	4.18E+02	**1.02E+03**	**5.85E+01**	1.07E+03	1.44E+02
F6	1.15E+07	1.27E+07	1.30E+06	4.98E+06	1.03E+07	1.98E+07	1.82E+09	8.74E+08	1.15E+06	6.30E+05	**6.76E+03**	**4.28E+03**
F7	2.23E+03	7.43E+01	2.12E+03	4.04E+01	2.17E+03	3.79E+01	2.21E+03	4.04E+01	2.10E+03	1.54E+01	**2.05E+03**	**1.42E+01**
F8	2.31E+03	8.05E+01	2.29E+03	7.05E+01	2.29E+03	5.89E+01	2.39E+03	1.15E+02	2.24E+03	4.33E+00	**2.23E+03**	**3.36E+00**
F9	2.63E+03	5.01E+01	2.55E+03	5.13E+01	2.59E+03	4.50E+01	3.26E+03	3.00E+02	2.49E+03	2.30E+00	**2.48E+03**	**1.21E+00**
F10	4.84E+03	1.41E+03	4.27E+03	1.15E+03	3.02E+03	1.01E+03	5.89E+03	1.50E+03	2.58E+03	3.24E+02	**2.57E+03**	**2.52E+02**
F11	4.50E+03	6.04E+02	4.63E+03	1.07E+03	5.59E+03	7.36E+02	8.44E+03	8.91E+02	3.22E+03	**7.32E+01**	**2.95E+03**	1.56E+02
F12	3.10E+03	1.22E+02	3.05E+03	7.64E+01	3.03E+03	6.04E+01	3.52E+03	2.09E+02	3.05E+03	**1.67E+01**	**2.99E+03**	3.19E+01

Table 4. Algorithm comparison results. Significant values are the best results in the comparison of algorithm performance.

Parameter	Value
$z_{terrain}$	Calculate based on terrain
z_{max}	50 m
$\theta_{\max}$	45°
$\phi_{\max}$	45°
d_{safe}	Calculate based on obstacle radius

Table 5. Path planning parameters.

		Start point	End point
Mountainous terrain	Terrain 1	(20,20,5)	(280,280,5)
	Terrain 2	(20,20,5)	(280,280,5)
Urban terrain	Terrain 1	(20,20,5)	(280,280,5)
	Terrain 2	(20,20,5)	(280,280,5)

Table 6. Terrain parameter 1.

The simulation will be run 50 times, and the optimal value, average value, and standard deviation will serve as the performance evaluation criteria for the algorithms.

For mountainous terrain:

From Fig. 9 it is evident that the ICPO trajectory is smoother and lower, while other algorithms have not found the optimal path or achieved the optimal cost path. From Fig. 10, it can be seen that some algorithms often fall into local optima in the early stages. Comparing the convergence curves, ICPO enhances the later development phase while maintaining the initial search intensity, thus improving convergence efficiency. Compared to other algorithms, ICPO is more stable and capable of finding the optimal value. These algorithms converge quickly in the early stages but often fall into local optima later on. ICPO demonstrates high efficiency in UAV delivery path planning in complex mountainous environments. Its ability to navigate such challenging terrain while maintaining optimal convergence highlights its robustness.

From Table 8, ICPO consistently outperforms the other algorithms across both terrains, achieving the best values of 778.1775 on Terrain 1 and 954.0118 on Terrain 2. Its mean values are 1021.7682 and 990.5413 with relatively low standard deviations, indicating stability and effectiveness in finding optimal routes. HISOS-SCPSO and HGWODE, show strong early performance but tend to get trapped in local optima and have higher variability. However, CAOSA, MODMA, and the unoptimized CPO do not reach ICPO's level of performance. Overall, ICPO ranks first, demonstrating superior capability in UAV delivery path planning in complex mountainous environments.

For urban terrain:

From Fig. 11 it is evident that the ICPO trajectory is smoother and lower, while other algorithms have not found the optimal path or achieved the optimal cost path. From Fig. 12, it can be seen that some algorithms often fall into local optima in the early stages. Comparing the convergence curves, ICPO enhances the later development phase while maintaining the initial search intensity, thus improving convergence efficiency.

	Mountainous terrain		Urban terrain	
	Terrain 1	Terrain 2	Terrain 1	Terrain 2
Obstacles center	[140, 50] [80, 70] [20, 90] [240, 110] [60, 130] [180, 150] [200, 70] [120, 190] [240, 210] [60, 230] [140, 180] [150, 200] [160, 200] [190, 210] [210, 190] [200, 140] [40, 140] [100, 120] [100, 80] [80, 200]	[140, 50] [65, 45] [30, 150] [220, 180] [60, 110] [180, 130] [250, 90] [80, 220] [210, 250] [120, 140] [50, 180] [190, 60] [140, 180] [220, 280] [100, 180] [90, 90] [66, 136] [190, 190] [130, 110] [180, 140]	[140, 50] [70, 170] [120, 150] [260, 190] [140, 210] [200, 170] [220, 270] [100, 250] [170, 130] [230, 230] [180, 180] [150, 90] [240, 150] [80, 220] [75, 80] [185, 230] [90, 140] [135, 170] [250, 100] [110, 200]	[140, 50] [80, 180] [130, 150] [270, 200] [150, 220] [210, 180] [230, 280] [110, 260] [180, 140] [240, 240] [190, 190] [160, 100] [250, 160] [90, 230] [85, 90] [195, 240] [100, 150] [145, 180] [260, 110] [120, 210]
Obstacles parameters (height, cross slope, longitudinal slope or height, radius)	(50, 20, 20) (25, 20, 15) (35, 20, 25) (30, 20, 10) (25, 15, 15) (18, 15, 10) (20, 25, 25) (35, 10, 15) (38, 15, 20) (32, 20, 20) (30, 30, 25) (35, 5, 10) (40, 10, 15) (42, 15, 25) (40, 10, 10) (48, 25, 15) (48, 20, 10) (40, 15, 25) (32, 10, 20) (50, 35, 25)	(50, 20, 20) (30, 22, 17) (40, 22, 27) (35, 22, 12) (50, 37, 37) (22, 17, 12) (25, 27, 27) (40, 12, 17) (42, 17, 22) (35, 22, 22) (33, 32, 27) (40, 7, 12) (35, 12, 17) (37, 17, 27) (35, 12, 12) (30, 27, 17) (30, 22, 12) (43, 17, 27) (35, 12, 22) (55, 37, 27)	(50, 20) (40, 15) (50, 10) (50, 10) (25, 15) (20, 10) (30, 15) (45, 15) (50, 15) (40, 10) (30, 12) (45, 18) (30, 14) (20, 16) (55, 10) (30, 16) (30, 14) (40, 12) (45, 10) (30, 15)	(50, 20) (40, 18) (50, 12) (50, 12) (25, 18) (20, 12) (30, 18) (45, 18) (50, 18) (40, 12) (30, 15) (45, 20) (30, 16) (20, 18) (55, 12) (30, 18) (30, 16) (40, 15) (45, 12) (30, 18)

Table 7. Terrain parameter 2.

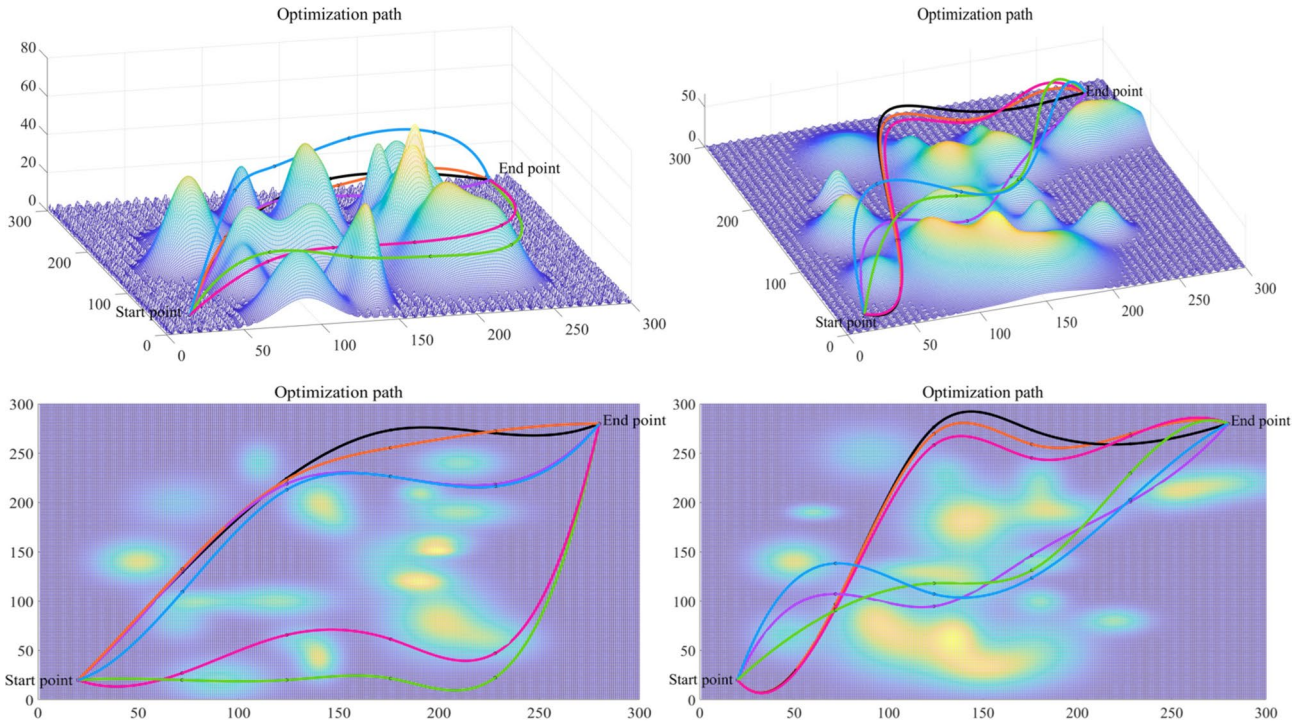

Fig. 9. Comparison of UAV Path.

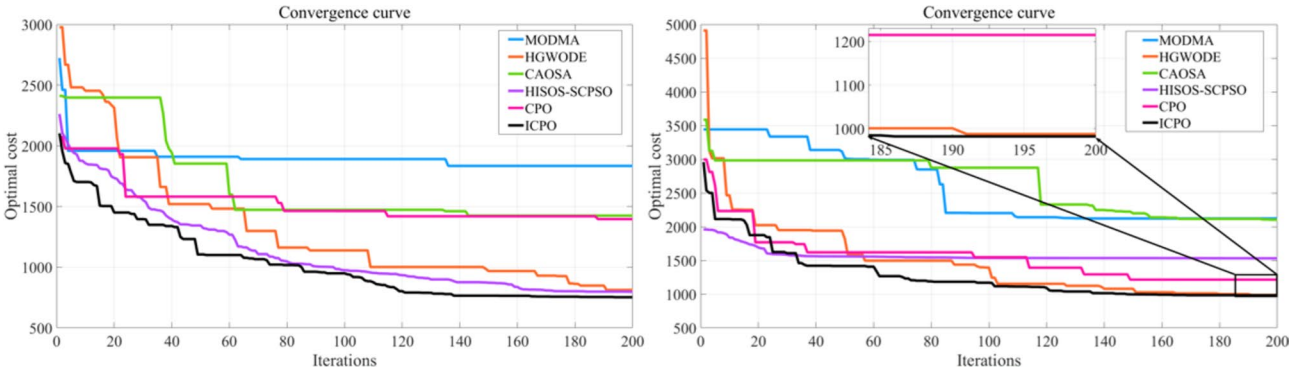

Fig. 10. Comparison of convergence curves.

		Terrain 1	Terrain 2	Best Rank
MODMA	Best	1543.4667	1962.5622	6
	Mean	1787.71	2322.0208	
	Std	138.3065	477.9647	
	Time (s)	52.0658	46.4426	
HGWODE	Best	792.8862	972.3973	3
	Mean	960.8046	998.4625	
	Std	192.7732	40.6152	
	Time (s)	17.1352	18.042	
CAOSA	Best	997.7967	1193.8264	4
	Mean	1272.9643	1684.2112	
	Std	143.1121	253.2906	
	Time (s)	10.3176	10.6027	
HISOS-SCPSO	Best	788.5011	955.3776	2
	Mean	1236.262	1444.3564	
	Std	276.9461	328.7374	
	Time (s)	26.7883	28.1941	
CPO	Best	1140.3637	1063.8942	5
	Mean	1311.8547	1244.8399	
	Std	72.9508	103.5488	
	Time (s)	4.1338	4.2789	
ICPO	Best	778.1775	954.0118	**1**
	Mean	1021.7682	990.5413	
	Std	168.9703	43.1635	
	Time (s)	4.7221	6.6796	

Table 8. Comparison of algorithm performance after 50 repeated runs.

Compared to other algorithms, ICPO is more stable and capable of finding the optimal value. These algorithms converge quickly in the early stages but often fall into local optima later on. ICPO demonstrates high efficiency in UAV delivery path planning in complex urban environments. Its ability to navigate such challenging terrain while maintaining optimal convergence highlights its robustness.

From Table 9, ICPO consistently outperforms the other algorithms across both terrains, achieving the best values of 366.2789 on Terrain 1 and 910.1682 on Terrain 2, ranking first. Its mean values are 821.3689 and 1244.8609 with relatively low standard deviations, indicating stability and effectiveness in finding optimal routes. HISOS-SCPSO, HGWODE, CAOSA, MODMA, and CPO, do not reach ICPO's level of performance, showing higher best values, mean values, and standard deviations. Overall, ICPO demonstrates superior capability in UAV delivery path planning in complex urban environments.

In summary, ICPO demonstrates significant advantages in UAV delivery path planning across various terrains. Comparative analysis shows that ICPO's trajectories consistently approach optimal values, showcasing its robustness and stability. The convergence curves indicate that ICPO not only maintains initial search intensity but also enhances later development stages, thereby improving convergence efficiency. Notably, ICPO outperforms other algorithms across all terrains. While HISOS-SCPSO and HGWODE exhibit strong early performance, they tend to fall into local optima in later stages. Therefore, ICPO excels in UAV delivery path planning in complex

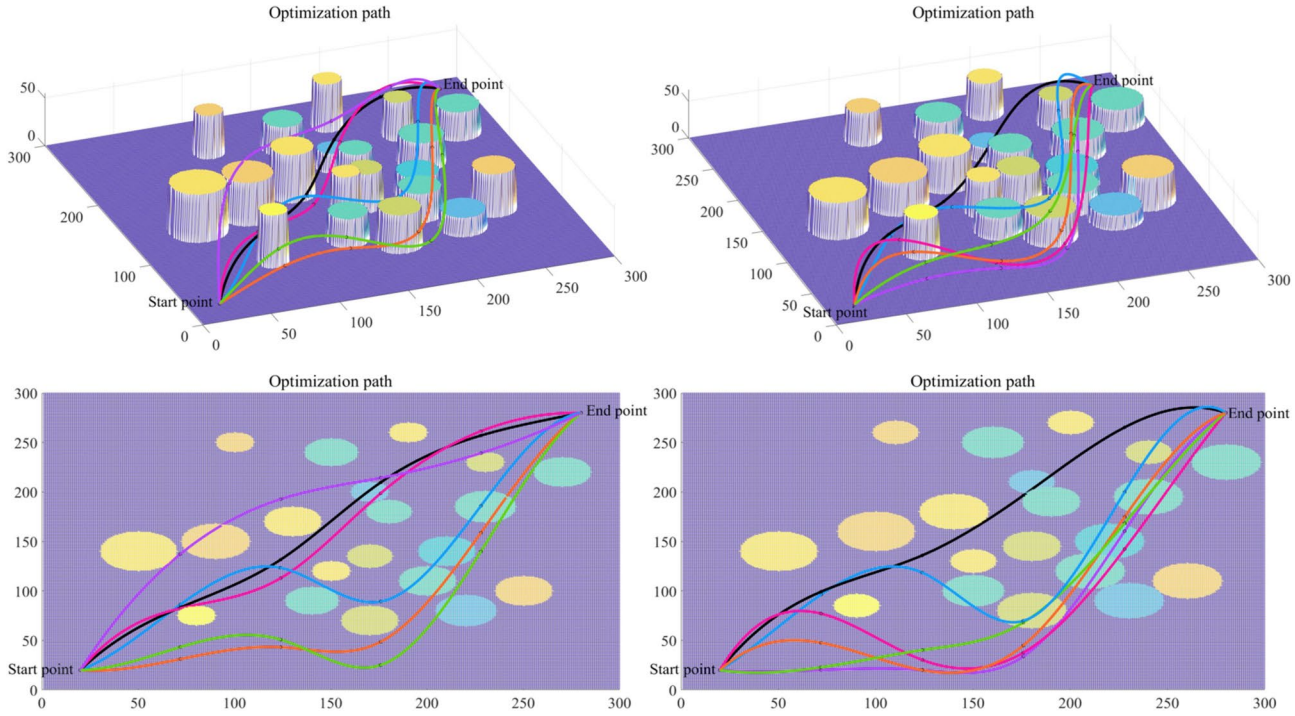

Fig. 11. Comparison of UAV Path.

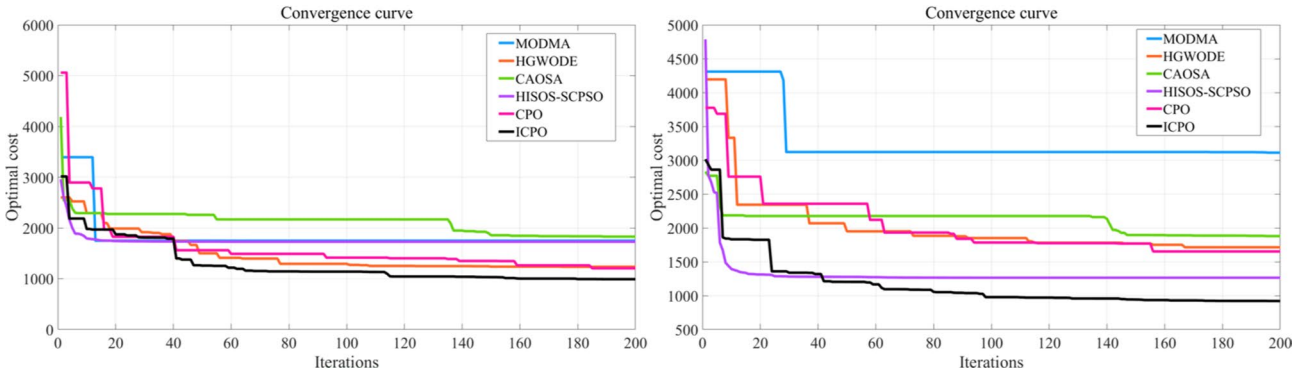

Fig. 12. Comparison of Convergence Curves.

environments, maintaining optimal performance and consistency throughout the optimization process. This makes ICPO particularly effective for applications requiring precise and reliable route optimization in challenging terrains.

Conclusion

Given the increasing demand for efficient and reliable delivery systems in challenging environments, this study developed an Improved Crowned Pig Optimizer (ICPO) for UAV path planning in complex environments. Inspired by the unique defensive behaviors of porcupines, the study proposes a vision and hearing collaborative perspective. By adopting an audiovisual interactive defense mechanism, it improves the original framework where vision is the primary defense means and hearing is the secondary, addressing the critical issue of slow early convergence in traditional CPO. Additionally, inspired by porcupines' balanced use of their defense mechanisms at various locations, the study adopts a good point set population initialization strategy to increase population diversity and removes the population reduction mechanism based on the average distribution characteristics of the population. To address the issue of traditional CPO easily falling into local optima in the later stages and inspired by porcupines' refined defensive mechanisms to protect themselves, the study proposes a novel periodic retreat strategy to improve position updates and help ICPO escape local optima.

Comparative analysis shows that ICPO not only consistently achieves near-optimal values on the test set but also demonstrates robustness and stability in UAV path planning applications. In complex mountainous terrain, ICPO achieved optimal values of 778.1775 and 954.0118; in urban terrain, 366.2789 and 910.1682 and ranked first among the compared algorithms, it has demonstrated its efficiency and dependability in drone delivery route planning.

		Terrain 1	Terrain 2	Best Rank
MODMA	Best	1389.7489	1640.1152	6
	Mean	1733.2538	1972.0262	
	Std	237.9423	195.6743	
	Time (s)	31.51	30.2143	
HGWODE	Best	977.9323	1265.811	3
	Mean	1201.737	1446.6392	
	Std	82.8609	125.7724	
	Time (s)	14.9854	14.4587	
CAOSA	Best	1273.7542	1349.1382	5
	Mean	1574.7066	1714.2137	
	Std	216.8093	153.7684	
	Time (s)	8.8616	8.7456	
HISOS-SCPSO	Best	450.1434	1244.5138	2
	Mean	1289.7589	1446.9648	
	Std	476.3155	223.1161	
	Time (s)	22.7995	22.8503	
CPO	Best	987.003	1257.7766	4
	Mean	1189.3776	1422.957	
	Std	111.7632	83.2302	
	Time (s)	3.3244	3.0312	
ICPO	Best	366.2789	910.1682	**1**
	Mean	821.3689	1244.8609	
	Std	247.298	183.3681	
	Time (s)	5.3793	4.9982	

Table 9. Comparison of algorithm performance after 50 repeated runs.

ICPO marks a significant advancement in UAV path planning, particularly in complex environments. Future research could focus on further enhancing the adaptability and scalability of ICPO to cater to even more diverse and dynamic scenarios. Additionally, exploring the integration of ICPO with real-time environmental data and machine learning techniques could unlock new potentials for autonomous systems, ensuring even higher efficiency and reliability in various applications such as disaster response, logistics, and surveillance. The continuous improvement and real-world testing of ICPO will be crucial in advancing the field of autonomous navigation and optimization algorithms.

Data availability

Since the algorithm code is a core component and is currently being used during the research period, it can be obtained from the corresponding author Z.J. upon reasonable request.

Received: 18 July 2024; Accepted: 28 August 2024
Published online: 03 September 2024

References

1. Raghunatha, A. et al. Addressing the emergence of drones—A policy development framework for regional drone transportation systems. *Transp. Res. Interdiscip. Perspect.* **18**, 100795. https://doi.org/10.1016/j.trip.2023.100795 (2023).
2. Kellermann, R. et al. Drones for parcel and passenger transportation: A literature review. *Transp. Res. Interdiscip. Perspect.* **4**, 100088. https://doi.org/10.1016/j.trip.2019.100088 (2020).
3. Niu, B. et al. Drone logistics' resilient development: Impacts of consumer choice, competition, and regulation. *Transp. Res. Part A Policy Pract.* **185**, 104126. https://doi.org/10.1016/j.tra.2024.104126 (2024).
4. Sun, X. et al. UAV-rider coordinated dispatching for the on-demand delivery service provider. *Transp. Res. Part E Logist. Transp. Rev.* **186**, 103571. https://doi.org/10.1016/j.tre.2024.103571 (2024).
5. Frederiksen, M. H. et al. Citizen visions of drone uses and impacts in 2057: Far-future insights for policy decision-makers. *Technol. Forecast. Soc. Chang.* **204**, 123438. https://doi.org/10.1016/j.techfore.2024.123438 (2024).
6. Ait Saadi, A. et al. UAV path planning using optimization approaches: A survey. *Arch. Comput. Methods Eng.* **29**, 4233–4284. https://doi.org/10.1007/s11831-022-09742-7 (2022).
7. Boulares, M. et al. UAV path planning algorithm based on Deep Q-Learning to search for a floating lost target in the ocean. *Robot. Auton. Syst.* https://doi.org/10.1016/j.robot.2024.104730 (2024).
8. Lee, G. et al. Real-time path planning of controllable UAV by subgoals using goal-conditioned reinforcement learning. *Appl. Soft Comput.* **146**, 110660. https://doi.org/10.1016/j.asoc.2023.110660 (2023).
9. Zhu, X. et al. Path planning of multi-UAVs based on deep Q-network for energy-efficient data collection in UAVs-assisted IoT. *Veh. Commun.* **36**, 100491. https://doi.org/10.1016/j.vehcom.2022.100491 (2022).

10. Lee, M. H. & Moon, J. Deep reinforcement learning-based model-free path planning and collision avoidance for UAVs: A soft actor–critic with hindsight experience replay approach. *ICT Express* **9**(3), 403–408. https://doi.org/10.1016/j.icte.2023.403408 (2023).
11. Swain, S. et al. A reinforcement learning-based cluster routing scheme with dynamic path planning for multi-UAV network. *Veh. Commun.* **41**, 100605. https://doi.org/10.1016/j.vehcom.2023.100605 (2023).
12. Huang, T. et al. Density gradient-RRT: An improved rapidly exploring random tree algorithm for UAV path planning. *Expert Syst. Appl.* **252**, 124121. https://doi.org/10.1016/j.eswa.2024.124121 (2024).
13. Guo, J. et al. Feedback RRT* algorithm for UAV path planning in a hostile environment. *Comput. Ind. Eng.* **174**, 108771. https://doi.org/10.1016/j.cie.2022.108771 (2022).
14. Feng, Z. et al. DBVS-APF-RRT*: A global path planning algorithm with ultra-high speed generation of initial paths and high optimal path quality. *Expert Syst. Appl.* https://doi.org/10.1016/j.eswa.2024.123571 (2024).
15. Rao, J. et al. Path planning for dual UAVs cooperative suspension transport based on artificial potential field-A* algorithm. *Knowl.-Based Syst.* **277**, 110797. https://doi.org/10.1016/j.knosys.2023.110797 (2023).
16. Guo, J. et al. ICRA: An intelligent clustering routing approach for UAV ad hoc networks. *IEEE Trans. Intell. Transp. Syst.* **24**(2), 2447–2460. https://doi.org/10.1109/TITS.2022.3145857 (2022).
17. Liu, Z. et al. Lightweight trustworthy message exchange in unmanned aerial vehicle networks. *IEEE Trans. Intell. Transp. Syst.* **24**(2), 2144–2157. https://doi.org/10.1109/TITS.2021.3136304 (2021).
18. Liu, Y. et al. A fast formation obstacle avoidance algorithm for clustered UAVs based on artificial potential field. *Aerospace Sci. Technol.* https://doi.org/10.1016/j.ast.2024.108974 (2024).
19. Chen, Y. et al. Modified central force optimization (MCFO) algorithm for 3D UAV path planning. *Neurocomputing* **171**, 878–888. https://doi.org/10.1016/j.neucom.2015.07.151 (2016).
20. Zhang, X. et al. A novel phase angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism applied to UAV path planning. *Appl. Soft Comput.* **70**, 371–388. https://doi.org/10.1016/j.asoc.2018.06.019 (2018).
21. Qu, C. et al. A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning. *Appl. Soft Comput.* **89**, 106099. https://doi.org/10.1016/j.asoc.2020.106099 (2020).
22. Avcu, M. E. et al. A WOA-based path planning approach for UAVs to avoid collisions in cluttered areas. In *Handbook of Whale Optimization Algorithm*, 449–461 https://doi.org/10.1016/B978-0-12-824457-2.00013-1 (Academic Press, 2024).
23. Yu, X. et al. A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios. *Knowl.-Based Syst.* **204**, 106209. https://doi.org/10.1016/j.knosys.2020.106209 (2020).
24. Huang, C. et al. Adaptive cylinder vector particle swarm optimization with differential evolution for UAV path planning. *Eng. Appl. Artif. Intell.* **121**, 105942. https://doi.org/10.1016/j.engappai.2023.105942 (2023).
25. Chai, X. et al. Multi-strategy fusion differential evolution algorithm for UAV path planning in complex environment. *Aerospace Sci. Technol.* **121**, 107287. https://doi.org/10.1016/j.ast.2022.107287 (2022).
26. Hu, G. et al. SaCHBA_PDN: Modified honey badger algorithm with multi-strategy for UAV path planning. *Expert Syst. Appl.* **223**, 119941. https://doi.org/10.1016/j.eswa.2023.119941 (2023).
27. Zhang, C. et al. A novel UAV path planning approach: Heuristic crossing search and rescue optimization algorithm. *Expert Syst. Appl.* **215**, 119243. https://doi.org/10.1016/j.eswa.2022.119243 (2023).
28. Karthik, K. & Balasubramanian, C. Improved Green Anaconda Optimization Algorithm-based Coverage Path Planning Mechanism for heterogeneous unmanned aerial vehicles. *Sustain. Comput. Inform. Syst.* **42**, 100961. https://doi.org/10.1016/j.suscom.2024.100961 (2024).
29. Qadir, Z. et al. Autonomous UAV path-planning optimization using metaheuristic approach for predisaster assessment. *IEEE Internet Things J.* **9**(14), 12505–12514. https://doi.org/10.1109/JIOT.2021.3137331 (2021).
30. Fan, C. et al. Flexible job shop scheduling with stochastic machine breakdowns by an improved tuna swarm optimization algorithm. *J. Manuf. Syst.* **74**, 180–197. https://doi.org/10.1016/j.jmsy.2024.01.009 (2024).
31. Abdel-Basset, M. et al. A Kepler optimization algorithm improved using a novel Lévy-Normal mechanism for optimal parameters selection of proton exchange membrane fuel cells: A comparative study. *Energy Rep.* **11**, 6109–6125. https://doi.org/10.1016/j.egyr.2024.06.005 (2024).
32. Pan, K. et al. Optimal scheduling of electric vehicle ordered charging and discharging based on improved gravitational search and particle swarm optimization algorithm. *Int. J. Electr. Power Energy Syst.* **157**, 109766. https://doi.org/10.1016/j.ijepes.2024.109766 (2024).
33. Pan, C. et al. Optimization of rolling bearing dynamic model based on improved golden jackal optimization algorithm and sensitive feature fusion. *Mech. Syst. Signal Process.* **204**, 110845. https://doi.org/10.1016/j.ymssp.2023.110845 (2023).
34. Krishnan, H. et al. Parameter identification of solar cells using improved Archimedes Optimization Algorithm. *Optik* **295**, 171465. https://doi.org/10.1016/j.ijleo.2023.171465 (2023).
35. SeyedGarmroudi, S. et al. Improved Pelican optimization algorithm for solving load dispatch problems. *Energy* **289**, 129811. https://doi.org/10.1016/j.energy.2024.129811 (2024).
36. Pan, X. et al. Improved artificial bee colony algorithm based on two-dimensional queue structure for complex optimization problems. *Alex. Eng. J.* **86**, 669–679. https://doi.org/10.1016/j.aej.2024.01.001 (2024).
37. Abdel-Basset, M. et al. Crested Porcupine Optimizer: A new nature-inspired metaheuristic. *Knowl.-Based Syst.* **284**, 111257. https://doi.org/10.1016/j.knosys.2023.111257 (2024).
38. He, G. & Lu, X. L. Good point set and double attractors based-QPSO and application in portfolio with transaction fee and financing cost. *Expert Syst. Appl.* **209**, 118339. https://doi.org/10.1016/j.eswa.2022.118339 (2022).
39. Cheng, M. Y. & Prayogo, D. Symbiotic organisms search: A new metaheuristic optimization algorithm. *Comput. Struct.* **139**, 98–112. https://doi.org/10.1016/j.compstruc.2014.03.007 (2014).
40. 2022 IEEE Congress on Evolutionary Computation (CEC). In *2022 IEEE Congress on Evolutionary Computation (CEC), Padua, Italy*, 1, https://doi.org/10.1109/CEC55065.2022.9870329 (2022).
41. Mirjalili, S. & Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008 (2016).
42. Lian, J. et al. Parrot optimizer: Algorithm and applications to medical problems. *Comput. Biol. Med.* **172**, 108064. https://doi.org/10.1016/j.compbiomed.2024.108064 (2024).
43. Wang, J. et al. Black-winged kite algorithm: A nature-inspired meta-heuristic for solving benchmark functions and engineering problems. *Artif. Intell. Rev.* **57**(4), 98. https://doi.org/10.1007/s10462-024-10723-4 (2024).
44. Dehghani, M. et al. Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. *Knowl.-Based Syst.* **259**, 110011. https://doi.org/10.1016/j.knosys.2022.110011 (2023).
45. Wang, X. et al. Modified mayfly algorithm for UAV path planning. *Drones* **6**(5), 134. https://doi.org/10.3390/drones6050134 (2022).
46. Yu, X. et al. A hybrid algorithm based on grey wolf optimizer and differential evolution for UAV path planning. *Expert Syst. Appl.* **215**, 119327. https://doi.org/10.1016/j.eswa.2022.119327 (2023).
47. Ait-Saadi, A. et al. A novel hybrid chaotic Aquila optimization algorithm with simulated annealing for unmanned aerial vehicles path planning. *Comput. Electr. Eng.* **104**, 108461. https://doi.org/10.1016/j.compeleceng.2022.108461 (2022).
48. Xiong, T. et al. A hybrid improved symbiotic organisms search and sine-cosine particle swarm optimization method for drone 3D path planning. *Drones* **7**(10), 633. https://doi.org/10.3390/drones7100633 (2023).

Acknowledgements
This work is supported by the National Natural Science Foundation/Youth Science Foundation of China (Grant No. 42101339). Furthermore, we would like to show our greatest appreciation to anonymous reviewers, editor, Ningbo University and those who have helped to contribute to this paper writing. The entire experimental process was completed using Matlab 2023b.

Author contributions
Conceptualization, Z. J. and H. L.; methodology, S. L. and Z. J.; software, S. L.; validation, H. L.; formal analysis, S. L. and H. L.; investigation, H. L.; resources, S. L.; data curation, S. L.; writing—original draft preparation, S. L.; writing—review and editing, Z. J.; visualization, S. L. and H. L.; supervision, Z. J.; project administration, Z. J.; funding acquisition, Z. J.. All authors have read and agreed to the published version of the manuscript.

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to Z.J.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.