Reinforcement Learning-Augmented Particle Swarm Optimization with Orthogonal Perturbation Operator for Multi-UAV Path Planning

Duohua Wang, Weifeng Gao, Zhengjun Wang, Hong Li, Jin Xie

Abstract

Multi-unmanned aerial vehicle (Multi-UAV) path planning in complex three-dimensional (3D) environments is a challenging task due to high-dimensional search spaces, multiple constraints, and numerous local optima. To address these issues, this paper proposes a particle swarm optimization-based orthogonal perturbation operator algorithm (OPOPSO). The proposed algorithm, which introduces an orthogonal perturbation operator to break the rigid update pattern of traditional particle swarm optimization, can enhance global exploration and alleviate premature convergence. A dynamic topology reorganization mechanism is designed to establish randomized interaction topologies and select guiding exemplars for each particle, which enhances population diversity. Furthermore, an adaptive parameter control framework based on reinforcement learning is proposed. This framework integrates a Dueling Double Deep Q-Network with a dual reward mechanism to alleviate the sparse reward problem. Experiments are conducted on the CEC2017 benchmark suite, as well as on path planning simulations using real-world digital elevation model (DEM) maps derived from Australian LiDAR data. Experimental results illustrate that OPOPSO achieves competitive performance against 16 advanced optimization algorithms, including seven PSO variants, four differential evolution variants, and five other metaheuristics algorithms, on the benchmark suite, while also maintaining superior solution quality and scalability in complex 3D path planning tasks.

Index Terms

Multi-UAV path planning, particle swarm optimization, reinforcement learning, global optimization, orthogonal perturbation operator.

I. Introduction

UNMANNED Aerial Vehicles (UAVs) offer significant advantages, including cost-effectiveness and operational flexibility, enabling them to replace human operators in high-risk environments. Currently, UAVs are extensively employed in domains including disaster rescue [1], target tracking [2], freight transportation [3], and military missions [4]. However, a single UAV has inherent limitations in payload capacity and mission efficiency, which hinder its ability to meet the requirements of complex environments. Therefore, multi-UAV cooperative mission execution has become a field of significant interest. A key challenge is safe trajectory planning in complex threat environments. This involves planning safe flight trajectories from starting points to target points for the UAVs within a prescribed mission area. The paths must satisfy multiple constraints, including terrain obstacle avoidance, inter-UAV separation maintenance and environmental threat avoidance. The challenge is further compounded when UAVs share common start and target points, a typical setup in cooperative missions, which increases collision risks and imposes stricter separation requirements. Essentially, this problem can be categorized as a multi-constrained path planning task, whose solution is crucial for strengthening the cooperative capability and mission reliability of multi-UAV systems.

The coordination of UAV operations, particularly in critical missions requiring precise timing, sequential execution, and collision avoidance, poses significant challenges [5]. Traditional path planning methods, including but not limited to the A* algorithm [6], and rapidly-exploring random tree (RRT) [7], have demonstrated efficiency and reliability in two-dimensional spaces. However, in complex three-dimensional (3D) environments the time complexity of these methods scales poorly, such that effective path planning becomes challenging.

Based on the limitations of traditional methods, intelligent optimization algorithms have become a research hotspot in multi-UAV path planning. These methods mainly include neural networks and evolutionary algorithms. Neural networks demand extensive training periods and exhibit limited generalization capabilities which constrains their effectiveness in dynamic or unpredictable environments [8].

In contrast, evolutionary algorithms are inspired by natural biological evolution mechanisms. They do not require precise mathematical formulations and can converge to optimal or high-quality suboptimal solutions within a reasonable time [9]. As one of the most prominent evolutionary algorithms, particle swarm optimization (PSO) draws inspiration from bird flocking behavior [10]. To date, PSO has attained significant success in the field of UAV path planning [4] [11] [12]. Compared with learning-based path search algorithms, PSO exhibits better adaptability and flexibility, while being easier to implement

This work was supported in part by the National Nature Science Foundation of China under Grant 62276202, in part by the Fundamental Research Funds for the Central Universities under Grant QTZX25074, in part by the Joint Funds of the Zhejiang Provincial Natural Science Foundation of China under Grant LHZY24A010005, and in part by National Key Laboratory of Science and Technology on Space Microwave under Grant HTKJ2024KL504008. (Corresponding authors: Weifeng Gao.)

Duohua Wang and Weifeng Gao are with the School of Mathematics and Statistics, Key Laboratory of Collaborative Intelligence Systems, Ministry of Education, Xidian University, Xi'an 710071, China (e-mail: 24071213204@stu.xidian.edu.cn; gaoweifeng2004@126.com).

and deploy [8]. It provides theoretical support and technical guidance for multi-UAV cooperative path planning in complex environments.

However, research on multi-UAV cooperative 3D path planning is still insufficient in the field of evolutionary algorithms. Most existing work focuses on either 2D multi-UAV coordination or 3D single-UAV planning. The few studies that do consider 3D multi-UAV scenarios assume that UAVs have distinct start and target points. This allows them to avoid the difficulty of coordinating multiple UAVs in both space and time when they share the same departure and arrival locations. As a result, maintaining safe separation becomes infeasible under heavily overlapping trajectories, and this high-risk scenario has yet to be systematically investigated. This paper addresses this gap by tackling cooperative 3D path planning under shared start and target points, where collision risk is highest. On the basis of the foregoing analysis, the core contributions of this work are summarized below:

1) An OPOPSO algorithm is proposed for multi-UAV 3D path planning. The proposed algorithm introduces an orthogonal perturbation operator, a dynamic topology reorganization mechanism and a dynamic population reduction strategy, which enhance its global exploration and local optimization capabilities in 3D environments. Among these mechanisms, the orthogonal perturbation operator alleviates the premature convergence of traditional PSO in complex terrains and reduces the length of the planned path.

2) An adaptive parameter control framework based on deep reinforcement learning (RL) is designed. It formalizes parameter adjustment of evolutionary algorithms as a Markov decision process and employs the Dueling Double Deep Q-Network (D3QN) for parameter selection. To address the problem of sparse rewards, a dual reward mechanism that combines intrinsic and extrinsic rewards is proposed. This mechanism stabilizes the learning of parameter strategies and improves the resilience and adaptability of the algorithm.

3) A 3D environment model with multiple obstacles is constructed using a 5-meter resolution grid digital elevation model (DEM) derived from Australian LiDAR data[1] This model provides the algorithm with environmental support that conforms to actual application scenarios.

The structure of this paper is outlined as follows. Section II reviews related work on path planning. The path planning problem model is formulated in Section III. Section IV presents the proposed OPOPSO algorithm. Section V reports experimental results on the CEC2017 benchmark suite and simulated path planning test sets, including comparisons with recently proposed PSO variants and other advanced evolutionary algorithms (EAs). Section VI concludes this paper.

II. Related Work

The path planning problem has been extensively studied, yielding numerous methods. Wang *et al.* [13] proposed the EBS-A* algorithm, which enhances the traditional A* algorithm through expanded distance, bidirectional search, and smoothing optimization for mobile robot path planning. This method not only enhances planning efficiency but also minimizes critical nodes and sharp turns. Fareh *et al.* [14] proposed the Sobel potential field (SPF) algorithm, which reduces computational load by focusing on obstacle regions through edge detection. Combined with PSO optimization and linear shortcut strategies, it balances speed and quality in UAV path planning. Chang *et al.* [15] developed an improved RRT algorithm that integrates skeleton extraction with a greedy algorithm. Together with minimum-snap trajectory optimization and proportional differential control, it enables accurate 3D trajectory tracking for UAVs. Although traditional methods exhibit favorable performance in discrete environments, their computational cost tends to escalate under continuous problems with multiple constraints.

In contrast, evolutionary algorithms show great potential in solving continuous, multi-constraint path planning problems. Yu *et al.* [16] proposed a hybrid algorithm HGWODE that integrates grey wolf optimization (GWO) with differential evolution (DE). To enhance exploitation, the algorithm modifies the position update equation of GWO, while a rank-based mutation strategy in DE balances exploration and exploitation. Shen *et al.* [17] proposed the ANSGA-III-PPS algorithm based on multi-stage constraint handling. It adopts a push-pull search framework for adaptive constraint processing and designs a preference-point mutation strategy to improve offspring generation. Yu *et al.* [18] proposed an enhanced dung beetle optimization algorithm (EDBO), which integrates an optimal value guidance strategy, a nonlinear adjustment factor, a boundary control scheme, and an improved foraging mechanism. These enhancements are designed to mitigate the slow convergence and low accuracy of the original DBO. Experiments on CEC benchmarks and UAV path planning demonstrate that EDBO outperforms competing metaheuristics in terms of accuracy, convergence speed, and stability. Huang *et al.* [4] proposed the ACVDEPSO algorithm, which integrates cylinder-surface vectors, adaptive parameters, and differential evolution operators. It simplifies the search by converting particle velocities into cylinder-surface vectors and generates challengers from historical global best solutions to escape from local optima. Cui *et al.* [19] proposed the fractional-order artificial bee colony algorithm (FOABC), which integrates the artificial bee colony algorithm (ABC) with fractional-order calculus to enhance local exploitation via the memory property in the onlooker phase and employs a DE-based strategy in the employed phase to balance exploration. Its effectiveness is validated on CEC2017 and robot path planning. Yu *et al.* [20] extended evolutionary optimization for UAV path planning by introducing a carbon-footprint efficiency ratio, along with adaptive population shrinking and carbon-budget termination, which reduces carbon emissions in 3D tasks with marginal accuracy loss. This green optimization approach is a promising avenue for

[1]The data of digital elevation model can be found at https://www.ga.gov.au/scientific-topics/national-location-information.

future work. While these algorithms perform well for single-UAV 3D planning, they become intractable in multi-UAV settings with shared start and target points due to the curse of dimensionality and tightened constraints.

In recent years, reinforcement learning has been widely hybridized with evolutionary algorithms. Yu and Luo [21] addressed 3D UAV path planning problem by proposing the RLMSCS algorithm. This algorithm introduces a reinforcement learning mechanism to design a multi-strategy search framework, where reinforcement learning selects appropriate search strategies and parameters. Cui *et al.* [22] proposed a reinforcement learning enhanced artificial bee colony algorithm (ABC_RL) to address the poor exploitation and slow convergence of the standard ABC algorithm. This algorithm integrates reinforcement learning to tune the number of dimensions in the employed-bee phase and designs two improved strategies to enhance search randomness. Wu *et al.* [23] proposed a Q-learning enhanced multi-strategy exponential distribution optimizer to enhance its exploitation capability. It enriches the exploration mechanism through differential strategies and adjusts parameters. Yang and Yu *et al.* [24] proposed a reinforced quantum Aquila Optimizer (RLQFAO) to address the premature convergence and poor adaptability of the standard Aquila Optimizer in complex 3D UAV path planning. This algorithm introduces a Q-learning-driven multi-strategy search framework, and enriches exploration capability through quantum rotation gates and moth-flame search operators, yielding improved convergence accuracy and robustness in multi-threat environments. The aforementioned evolutionary algorithms demonstrate superior convergence speed and adaptability in complex environments for single-UAV path planning. However, cooperative path planning for UAVs requires further investigation. While RL-enhanced EAs improve adaptability, relying on a single extrinsic reward leads to sparse supervision and unstable training. Additionally, Q-learning restricts state and action to discrete domains, limiting the algorithm's capability.

Unlike single-UAV path planning, multi-UAV path planning entails not only higher dimensionality but also additional constraints, including inter-UAV collision avoidance. The CL-DMSPSO algorithm was proposed by Xu *et al.* [25]. By considering communication constraints and collision-avoidance requirements, it plans cooperative paths for UAVs in 2D environment. Liu *et al.* [26] proposed a multi-UAV path planning algorithm that integrates a spatially refined voting mechanism with PSO. The particle swarm is divided into sub-swarms corresponding to individual UAVs, integrated with a voting-refinement and time-coordination mechanism. This approach achieves collision-free obstacle avoidance and reduces planning cost. Yu *et al.* [27] propose the Bounty Hunter Optimizer (BHO), which uses the Explorpolis rule and quantum probability rotation selection to curb center bias and preserve diversity. On the CEC2017 benchmark, BHO consistently outperforms advanced optimization algorithms. In constrained multi-UAV mobile edge computing and path planning (MEC-PP) system, it reduces cost by over 15% compared with advanced competitors in both small and large-scale scenarios Xing *et al.* [28] proposed the MSFO-QL algorithm, which integrates multi-swarm fruit fly optimization with Q-learning. The fruit fly population is partitioned into sub-swarms and Q-learning is used to select search modes, thereby expanding the search scope and enhancing efficiency for multi-UAV path planning. Yin *et al.* [29] proposed the HEA-PPO algorithm. This algorithm employs a multi-strategy operator pool and adaptive waypoint coding to balance path length and threat avoidance. Niu *et al.* [30] proposed the MEAEO-RL algorithm, fusing an improved artificial ecosystem optimization algorithm with reinforcement learning. By constructing a multi-strategy database to enhance population diversity and using Q-learning for strategy selection, the algorithm achieves a balance between efficiency and safety in cooperative 3D path planning. Existing multi-UAV planning studies have mostly been in 2D or simulated 3D settings, with little validation on real terrain. Furthermore, these studies do not consider the cooperative case with identical start and target points, which is the focus of our work.

III. Problem Description

The multi-UAV path planning problem considered in this paper involves determining, for each UAV, a feasible trajectory from the start point to the target point. The optimization aims to minimize the path length and flight risk, subject to terrain avoidance and no-fly zone constraints, as well as safe separation from ground obstacles and other UAVs. We address this problem in two parts. The first part concerns the representation of trajectories using waypoints and Bézier curves. The second part defines the evaluation function, which accounts for both the path cost and the degree of constraint satisfaction. These two components are detailed in the following subsections.

A. Path Representation

We describe the 3D trajectory of a UAV as an ordered sequence of waypoints $P = \{W_0, W_1, W_2, \ldots, W_n, W_{n+1}\}$ where W_0 and W_{n+1} represent the initial and final positions, respectively, and W_i represents the position at time t_i, such that the UAV reaches each waypoint at its corresponding time instant. Multi-UAV cooperative planning extends this representation to the joint optimization of multiple waypoint sequences $P_i = \{W_{i,0}, W_{i,1}, W_{i,2}, \ldots, W_{i,n}, W_{i,n+1}\}$, aiming to achieve optimal path performance via cooperative optimization. For path smoothing, we employ cubic Bézier curves to connect the waypoints. Given $n+2$ waypoints (including the start and end points) for UAV i, we construct $n+1$ cubic Bézier segments, denoted by $B_{i,j}(t)$ for $j = 0, 1, \ldots, n$ with $t \in [0, 1]$, such that $B_{i,j}(0) = W_{i,j}$ and $B_{i,j}(1) = W_{i,j+1} = B_{i,j+1}(0)$. To facilitate collision checking and cost evaluation, we insert $m-1$ sample points along each Bézier segment between consecutive waypoints. Let $w_{i,j}^k$ denote the k-th sample point on the j-th Bézier segment of UAV i, defined as follows:

$$w_{i,j}^{k} = B_{i,j}\left(\frac{k}{m}\right). \tag{1}$$

For notational simplicity, we set $W_{i,j} = w_{i,j}^{0}$ and introduce the global index $q = j \cdot m + k$, where $k \in \{0, 1, \ldots, m\}$. Consequently, the waypoints satisfy $W_{i,j+1} = w_{i,j}^{m} = w_{i,j+1}^{0} = w_{i,q}$ with $q = (j+1)m$.

B. Evaluation Function

The path planning problem entails finding the shortest flyable path for each UAV to minimize the path cost, subject to various constraints [25]. To handle these constraints, penalty function methods are widely used to transform constrained optimization problems into unconstrained ones [31]. In such methods, constraints are classified into two categories: hard constraints and soft constraints [32]. A solution is feasible if and only if all hard constraints are satisfied; any violation of a hard constraint renders the solution infeasible. When evaluating a solution, an infeasible solution is always considered inferior to a feasible one, regardless of its objective function value. Two solutions are compared according to their objective function values only when they are both feasible or both infeasible. Soft constraints, in contrast, are those we seek to satisfy as much as possible, while allowing a certain degree of violation. Although violating a soft constraint incurs a penalty in the objective function and thus degrades solution quality, the solution remains feasible. In this work, terrain and no-fly zone avoidance are imposed as hard constraints, since collisions with terrain or incursions into no-fly zones are unacceptable; any violation renders the path infeasible. In contrast, insufficient separation distances from no-fly zones, ground obstacles, and other UAVs only increase flight risk when falling below a safety threshold, without directly invalidating the path. Hence, such requirements are treated as soft constraints. Based on the above principles, the evaluation function for the multi-UAV path planning problem is defined as

$$F = f_l + \alpha\left(f_c + f_e + f_s + f_m\right), \tag{2}$$

where f_l, f_c, f_e, f_s, and f_m denote the path length cost, terrain collision cost, environmental threat cost, path segment cost, and separation maintenance cost, respectively. The formulation adopts a weighted sum approach to scalarize the multi-objective optimization problem, combining the primary path cost and the soft-constraint penalties into a single objective. The weighting parameter α balances the soft-constraint terms against the primary objective; a sufficiently large α penalizes soft-constraint violations with a large cost, thereby minimizing them. In this work, α is set to 100 to prioritize safety over path length minimization.

1) The Cost of Path Length: Path length is a fundamental component of the evaluation function, as shorter paths imply lower energy consumption. The total path length is then defined as

$$f_l = \sum_{i=1}^{N}\sum_{q=1}^{m(n+1)} \left\| w_{i,q} - w_{i,q-1} \right\|_2, \tag{3}$$

where N denotes the number of UAVs, and n denotes the number of waypoints other than the initial and final positions.

2) The Cost of Terrain Collision: The cost of terrain collision corresponds to the hard constraint on terrain and no-fly zone avoidance introduced above. For each sample point, the UAV is required to fly above the terrain and outside any no-fly zone. Let $h(w_{i,q})$ and $z(w_{i,q})$ denote the terrain elevation and the flight altitude at $w_{i,q}$, respectively, and let Ω_t denote the set of no-fly zones. The terrain collision cost is defined as follows:

$$f_c = \sum_{i=1}^{N}\sum_{q=0}^{m(n+1)} c(w_{i,q}), \tag{4}$$

where the penalty term $c(w_{i,q})$ is defined as

$$c(w_{i,q}) = \begin{cases} 0, & \text{if } z(w_{i,q}) > h(w_{i,q}) \text{ and } w_{i,q} \notin \Omega_t \\ M, & \text{otherwise} \end{cases}, \tag{5}$$

where M is a sufficiently large constant that makes any infeasible path prohibitively expensive in the overall cost.

3) The Cost of Environmental Threat: The cost of environmental threat accounts for collisions with terrain obstacles and no-fly zones. For terrain obstacles, each sample point $w_{i,q}$ defines a spherical safety region of radius r_s. Any terrain point falling within this sphere incurs a cost inversely proportional to the squared distance from the sample point. The terrain obstacle cost is defined as

$$f_t = \sum_{i=1}^{N}\sum_{q=0}^{m(n+1)}\sum_{l=1}^{n_t(w_{i,q})} \left(\frac{r_s}{r_l(w_{i,q})}\right)^2, \tag{6}$$

where $n_t(w_{i,q})$ is the number of terrain points within the safety sphere centered at $w_{i,q}$, and $r_l(w_{i,q})$ denotes the distance from the l-th such terrain point to $w_{i,q}$.

For no-fly zones, two geometric models are considered: cylindrical and hemispherical. Let n_c and n_h be the numbers of cylindrical and hemispherical no-fly zones, respectively. Let $d_c(w_{i,q}, l)$ denote the distance from $w_{i,q}$ to the l-th cylindrical no-fly zone, and $d_h(w_{i,q}, l)$ the distance to the l-th hemispherical no-fly zone. For each zone type, the cost is zero when the sample point is outside the safety radius r_s, and grows quadratically as the distance decreases within the radius. The cylindrical and hemispherical no-fly zone costs are defined as Eq.(7) and Eq.(8), respectively.

$$f_c(w_{i,q}, l) = \begin{cases} 0, & d_c(w_{i,q}, l) > r_s, \\ \left(\dfrac{r_s}{d_c(w_{i,q}, l)}\right)^2, & d_c(w_{i,q}, l) \le r_s, \end{cases} \tag{7}$$

$$f_h(w_{i,q}, l) = \begin{cases} 0, & d_h(w_{i,q}, l) > r_s, \\ \left(\dfrac{r_s}{d_h(w_{i,q}, l)}\right)^2, & d_h(w_{i,q}, l) \le r_s. \end{cases} \tag{8}$$

The total environmental threat cost is

$$f_e = f_t + \sum_{i=1}^{N} \sum_{q=0}^{m(n+1)} \left(\sum_{l=1}^{n_c} f_c(w_{i,q}, l) + \sum_{l=1}^{n_h} f_h(w_{i,q}, l) \right). \tag{9}$$

4) The Cost of Path Segments: Each waypoint $W_{i,j}$ represents the position of UAV i at time t_j. To respect the UAV's physical limits, a constraint is imposed on the distance between consecutive waypoints. The path segment cost penalizes violations of this constraint and is defined as

$$f_s = \sum_{i=1}^{N} \sum_{j=0}^{n} f_s(W_{i,j}), \tag{10}$$

where

$$f_s(W_{i,j}) = \begin{cases} d_{i,j} - r_d, & d_{i,j} \ge r_d, \\ 0, & d_{i,j} < r_d, \end{cases} \tag{11}$$

with $d_{i,j} = \|W_{i,j+1} - W_{i,j}\|_2$ denoting the distance between consecutive waypoints, and r_d being the minimum allowable distance threshold.

5) The Cost of Separation Maintenance: Unlike the preceding cost terms, which are computed independently for each UAV, the separation maintenance cost captures the inter-UAV coupling inherent in multi-UAV cooperative planning. To ensure flight safety, a minimum separation distance must be maintained between any two UAVs at all times. Under the temporal cooperation constraint introduced in Section III-A, all UAVs reach their corresponding waypoints simultaneously; consequently, the separation between UAVs is evaluated at each synchronized sample point. The separation maintenance cost is defined as

$$f_m = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{q=1}^{m(n+1)-1} f_m(w_{i,q}, w_{j,q}), \tag{12}$$

where

$$f_m(w_{i,q}, w_{j,q}) = \begin{cases} 0, & d_{ij}(q) \ge r_s, \\ \left(\dfrac{r_s}{d_{ij}(q)}\right)^2, & d_{ij}(q) < r_s, \end{cases} \tag{13}$$

with $d_{ij}(q) = \|w_{i,q} - w_{j,q}\|_2$ denotes the distance between UAV i and UAV j at the q-th synchronized sample point, and r_s is the minimum required separation distance.

IV. Proposed Method

This section presents the proposed OPOPSO algorithm. The overall framework consists of two main parts: the evolutionary search mechanism and an adaptive parameter control method based on reinforcement learning. These two parts are detailed in the following subsections.

A. Search Mechanism of OPOPSO

In traditional PSO, each particle in the swarm represents a candidate solution. The velocity and position of particle i at generation k are updated as

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 \left(pbest_i^k - X_i^k\right) + c_2 r_2 \left(gbest^k - X_i^k\right) \tag{14}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1}, \tag{15}$$

where $X_i^k = \{x_{i1}^k, x_{i2}^k, \cdots, x_{iD}^k\}$ and $V_i^k = \{v_{i1}^k, v_{i2}^k, \cdots, v_{iD}^k\}$ denote the position and velocity of the i-th particle at generation k, respectively. $pbest_i^k$ is the personal best position of particle i up to generation k, and $gbest^k$ is the global best position of the entire swarm. Equations (14) can be rewritten as

$$V_i^{k+1} = \omega_1 V_i^k + \omega_2 \delta_1 + \omega_3 \delta_2 \tag{16}$$

$$\delta_1 = E_1 - X_i^k \tag{17}$$

$$\delta_2 = E_2 - X_i^k, \tag{18}$$

where ω_1, ω_2 and ω_3 denote the weight coefficients. E_1 and E_2 are two exemplars that the current particle learns from, and δ_1, δ_2 are the difference vectors between the particle and each exemplar. However, PSO suffers from issues such as premature convergence to local optima, sensitivity to empirical parameter tuning, and rigid particle update mechanisms [33]. To address these issues, the OPOPSO algorithm is proposed. A fundamental characteristic of OPOPSO is that it uses the current particle position at each iteration, instead of the historical best position. The key components of the proposed approach are outlined as follows.

*1) **Orthogonal Perturbation Operator**:* In traditional PSO, particles are updated via a linear combination of the global best and personal best positions. This strategy converges rapidly early but often fixes the search direction later, making local optima escape difficult and limiting performance on complex problems [34]. To address this, an orthogonal perturbation operator is introduced, which adds random components orthogonal to the update direction. Given the difference vector δ_1 from Eq. (17), it is normalized to obtain the unit vector $\alpha_1 = \delta_1 / \|\delta_1\|$, and then a unit vector β_1 orthogonal to α_1 is constructed.

$$\beta_1 = \frac{r - (r \cdot \alpha_1)\alpha_1}{\|r - (r \cdot \alpha_1)\alpha_1\|}, \tag{19}$$

where r is a randomly generated unit vector. Applying Gram–Schmidt orthogonalization, we project r onto the orthogonal complement of α_1, yielding a vector β_1 for which $\alpha_1 \cdot \beta_1 = 0$. An adjustable angle parameter θ is then introduced to decompose the update into two orthogonal components:

$$\hat{\delta}_1 = \|\delta_1\| \left(\cos\theta \cdot \alpha_1 + \sin\theta \cdot \beta_1\right). \tag{20}$$

Following the same process, $\hat{\delta}_2$ can be obtained. After applying the orthogonal perturbation operator, the particle update formula is modified from Eq. (16) to Eq. (21).

$$V_i^{k+1} = \omega_1 V_i^k + \omega_2 \hat{\delta}_1 + \omega_3 \hat{\delta}_2. \tag{21}$$

This formulation adjusts the two directional components via trigonometric functions, striking a balance between the exploration and exploitation capabilities of the algorithm. It is noteworthy that when $\theta = 0$, Eq. (21) reduces to Eq. (16), meaning the traditional update is a special case of the proposed formulation. Fig. 1 illustrates the principle of the orthogonal perturbation operator, where vector addition follows the parallelogram law. The light blue dashed line represents the particle's movement governed by Eq. (16), which is prone to converging prematurely to local optima. In contrast, movement along the blue dashed line, updated via Eq. (21), escapes from local optima.

*2) **Dynamic Topology Reorganization Mechanism**:* In PSO, the selection of exemplar particles determines the effectiveness of particle updates. Traditional exemplar selection mechanisms face two core challenges. First, fixed topologies, such as ring topologies [35] or star topologies [36], restrict particle interaction ranges, reducing neighborhood diversity and causing particles to become trapped in local optima [37]. Second, fitness differences among exemplars are often neglected, resulting in homogeneous guidance directions that also lead to premature convergence. To overcome these issues, a dynamic topology reorganization mechanism is adopted, as illustrated in Fig. 2. The mechanism operates as follows:

- The reinforcement learning module outputs the optimal topology size in real time, which partitions the current swarm into stochastic topologies. Each particle interacts only with others in its own topology.
- Within each topology, particles are sorted by fitness, and those with better fitness than the current particle are selected as effective guides. This excludes inferior particles, allowing the update to focus on high-quality neighborhood information.
- Within each topology, two guiding particles, namely E_1 and E_2, are determined from the fitness ranking. E_1 is the best individual. E_2 is randomly selected from the remaining effective guides. This design preserves guidance effectiveness while introducing randomness to enhance exploration.

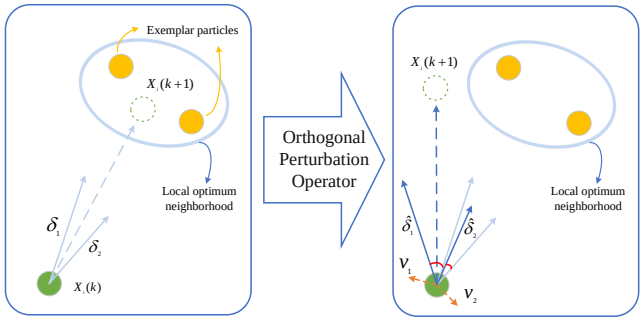


Fig. 1. Schematic diagram of orthogonal perturbation operator

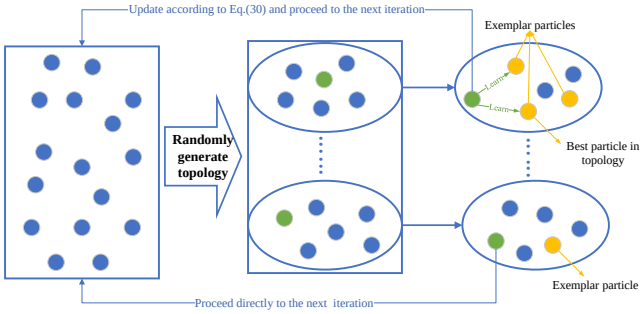


Fig. 2. Framework of topological neighborhood search mechanism

With this mechanism, particle i is updated as

$$V_i^{k+1} = R_1 V_i^k + R_2 \hat{\delta}_1 + \phi R_3 \hat{\delta}_2. \tag{22}$$

Where R_1, R_2, and R_3 are random vectors whose elements are uniformly distributed over $[0, 1]$, and $\phi \in [0, 0.4]$ is designed to increase population diversity. The vectors $\hat{\delta}_1$ and $\hat{\delta}_2$ are obtained from δ_1 and δ_2 via Eq. (20). If fewer than two individuals in the topology outperform the current particle, the update is skipped. This prevents performance degradation caused by inferior guidance and steers particles toward promising regions.

3) ***Dynamic Population Reduction Strategy****:* Traditional PSO employs a fixed population size, which fails to adapt to the varying exploration–exploitation requirements at different optimization stages. Researchers have integrated population reduction strategies into EAs to enhance performance [38], [39]. Inspired by these works, a linear population reduction strategy is adopted:

$$N = N_{\max} - (N_{\max} - N_{\min}) \frac{FES}{FES_{\max}}, \tag{23}$$

where N denotes the current population size, bounded by $N_{\max}$ and $N_{\min}$(set to $0.1 \times N_{\max}$ in our experiments). FES and $FES_{\max}$ denote the current and maximum numbers of function evaluations, respectively.

These three components form the OPOPSO search mechanism. However, the reliance on manual tuning of parameters such as topology size, ϕ, and θ remains a limitation, which is addressed by the adaptive control mechanism based on reinforcement learning described below.

B. Adaptive Parameter Control Based on Reinforcement Learning

The performance of PSO depends on its control parameters, yet conventional parameter tuning methods lack the adaptability required during the search process [40]. Reinforcement learning (RL) provides a natural framework for adaptive parameter control by learning a policy that maximizes cumulative reward based on state feedback [41]. In this work, a Dueling Double Deep Q-Network (D3QN) [2] is utilized to learn the adaptive parameter regulation policy for OPOPSO. D3QN integrates Double DQN [42] to mitigate Q-value overestimation, a dueling architecture [43] to separately estimate state value and action advantage, and prioritized experience replay [44] to focus training on more informative samples. This combination offers stable and efficient policy learning, making it well-suited for the high-dimensional and dynamic decision process of parameter control in evolutionary computation.

[2]For a detailed description of the D3QN architecture and its training mechanisms, the reader is referred to the supplementary material.

Algorithm 1 OPOPSO

Require: Maximum population size $N_{\max}$, Maximal number of fitness evaluations $FES_{\max}$, Trained D3QN agent

```
Randomly initialize population with size N = N_max;
Evaluate initial fitness of each particle;
FES = N_max;
while FES ≤ FES_max do
    Get adaptive parameters (topology size, φ and θ_max) from D3QN agent;
    Randomly select particles to form the topology;
    for each particle in the topology do
        if there are at least two dominators in the topology then
            Set the best and a random exemplar as the leading exemplars;
            Update the particle in accordance with Eq.(22);
            Evaluate the fitness and FES = FES + 1;
        end if
    end for
    N = N_max − (N_max − N_min) × (FES/FES_max);
end while
```

In this work, the parameter selection process is formalized as a Markov decision process defined as follows.

1) ***Environment:*** : For parameter selection problem, the environment is defined as the problem space of the optimization problem to be solved. The algorithm's process of solving this problem can be formalized as a reinforcement learning process.

2) ***State:*** : For the parameter selection problem in PSO, the state space is defined by three features: the global phase feature f_1, population distribution feature f_2, and search direction feature f_3.

To identify the algorithm's evolutionary stage, this paper adopts the ratio of FES to $FES_{\max}$ as the global phase feature. This feature reflects the evolutionary trend of the optimization process from global exploration to local convergence.

$$f_1 = \frac{FES}{FES_{\max}}. \tag{24}$$

To depict the overall distribution of the population, we construct the following population distribution feature:

$$f_2 = \frac{\alpha \cdot \text{mean}\left(\|X_i^k - \bar{X}^k\|_2\right)}{\beta\sqrt{D}}, \tag{25}$$

where $\bar{X}^k$ denotes to the average position of the population at generation k, and D is the problem dimension. α is constantly set to 5 in this paper, and β is obtained by

$$\beta = \frac{\sum_{i=1}^{D}(u_i - l_i)}{D}, \tag{26}$$

where u_i and l_i denote the minimum and maximum values of the i-th dimension, respectively. By using the parameter α and β, f_2 is constrained within $[0, 1]$. This feature is applicable to problems of different dimensions, where it characterizes the population distribution and helps determine whether the algorithm is trapped in local optima.

To address the issue that PSO is prone to local optima, this paper introduces a search direction feature based on cosine similarity. For each particle, the cosine similarity between its two difference vectors δ_1 and δ_2 (defined in Eq. (17)–(18)) is computed. The feature f_3 is then defined as

$$f_3 = \frac{1}{N}\sum_{i=1}^{N}\frac{\delta_1 \cdot \delta_2}{\|\delta_1\|\|\delta_2\|}. \tag{27}$$

When f_3 approaches 1, the two exemplars are likely in the same region, indicating a high risk of being trapped in local optima. This feature quantifies the convergence degree of search directions and provides a basis for strategy adjustment.

3) ***Action:*** : The action space is discrete and defined by three parameters: topology size, diversity parameter ϕ, and upper bound of angle parameter $\theta_{\max}$. In each iteration, a policy network selects these parameters based on the current state; the resulting action determines the parameters for the next iteration and is evaluated by environmental feedback.

4) ***Reward:*** : Two drawbacks arise in reward mechanism designs that combine traditional evolutionary algorithms with reinforcement learning. First, rewards based on the global optimum often become sparse in later stages, hindering the agent's ability to identify meaningful policy improvements. Second, the discrete binary reward scheme provides a fixed reward only for fitness improvements, leading to myopic good-or-bad evaluations. This rigid feedback limits exploration and may cause premature convergence.

To address these limitations, a compound reward mechanism is proposed that combines internal and external rewards. While maintaining local search efficiency, the mechanism introduces a global progress reward to help escape local optima.

The internal reward quantifies each particle's local contribution to the optimization progress. It is defined using normalized fitness differences as

$$p_i^k = f(X_i^k) - f(X_i^{k+1}) \tag{28}$$

$$r_1 = 2 \times \frac{\sum_i p_i^k - \min_i p_i^k}{\max_i p_i^k - \min_i p_i^k} - 1, \tag{29}$$

where $f(X_i^k)$ is the fitness of particle i at generation k and p_i^k is its fitness improvement. The normalization maps r_1 to $[-1, 1]$, ensuring stable incentives for local search. Unlike conventional discrete binary rewards, our continuous internal reward captures relative performance differences among particles, rather than absolute fitness gains. Even when all particles improve, their assigned rewards do not necessarily increase. This design prioritizes relative performance, preserving exploration capability and enabling trajectories with poor short-term progress to eventually reach better global optima.

Relying solely on r_1 is insufficient for efficient algorithm evolution, as the final performance of an evolutionary algorithm ultimately depends on the quality of the global optimum. Therefore, the global best particle is also considered, and a global progress reward is defined as

$$p_g^k = f(\hat{X}^k) - f(\hat{X}^{k+1}), \tag{30}$$

$$r_2 = 10 \times \frac{p_g^k}{\max_k p_g^k}, \tag{31}$$

where $f(\hat{X}^k)$ is the fitness of the global best solution at generation k, p_g^k is its improvement, and $\max_k p_g^k$ is the maximum improvement up to the current generation. The normalization scales r_2 to $[0, 10]$, ensuring reward stability and providing directional incentives for global convergence.

The total reward combines both components:

$$r = r_1 + r_2. \tag{32}$$

This compound reward function maintains local search efficiency while leveraging the long-term guidance of the global progress reward. Compared to traditional mechanisms, it offers significant improvements in both policy learning stability and solution optimality.

Algorithm 2 Training D3QN Agent for OPOPSO Parameter Selection

Require: D3QN network parameters θ, the number of epochs Q, maximum number of function evaluations $FES_{\max}$

```
Initialize network parameters θ;
for epoch = 1 to Q do
    Shuffle the order of training functions and generate a random permutation f_l;
    for each function f_i in f_l do
        Initialize PSO environment corresponding to function f_i to obtain the initial state s_0;
        Set r = 0, done = False, t = 0;
        while done = False do
            Select action a_t based on state s_t and the exploration strategy;
            Execute a_t to get the next state s_{t+1}, reward r_t and FES = FES + 1;
            if FES ≥ FES_max then
                done = True
            end if
            Store (s_t, a_t, r_t, s_{t+1}, done) into the experience buffer;
            Sample experiences from the buffer to update θ;
            r = r_t + γ_t r, s_t = s_{t+1}, t = t + 1;
        end while
        ε = max(0.98ε, 0.01);
    end for
end for
```

V. Experiments

In this section, we evaluate OPOPSO on two sets of experiments, namely the CEC2017 benchmark suite and a series of path planning simulations, to validate its effectiveness and efficiency. In the experiments for OPOPSO, the $N_{\max}$ = 1000. Three key parameters are dynamically selected by a reinforcement learning agent: θ, sampled uniformly from $[0, \theta_{\max}]$ where $\theta_{\max}$ ranges from 0 to 0.75 radians in increments of 0.05; the topology size, varying from 3 to 30 in steps of 1; and ϕ, ranging from 0.05 to 0.4 in steps of 0.05. During the training phase, the agent is trained on the 50-dimensional CEC2017 benchmark suite. Key

hyperparameters are set as follows: learning rate $\alpha = 0.005$, maximum population size N_{max} = 1000, training epochs Q = 120. The detailed training procedure for the D3QN agent is presented in Algorithm 2.

First, the OPOPSO algorithm is compared with seven established PSO variants and nine other advanced optimization algorithms on the 29 benchmark functions from CEC2017. The selected competing algorithms include seven representative PSO variants with distinct improvement mechanisms such as topology design, parameter adaptation, and learning strategies, four well-established DE variants, and five recent metaheuristics, some incorporating reinforcement learning. Subsequently, OPOPSO and its competitors are applied to a suite of multi-UAV 3D path planning simulation experiments. Relevant analyses are then performed to further validate the performance improvements of OPOPSO.

The parameter settings for PSO variants follow their original publications, namely, SDLSO [45], TPLSO [46], EAPSO [47], SDPSO [48], CL-DMSPSO [25], L-ICSO [38], and MFCPSO [49]. Their configurations are summarized in Table SI. Furthermore, nine other algorithms and their variants are configured as described in their respective original publications. These algorithms are TLDE [50], CJADE [51], GosDE [52], ISHACDE [53], DGTCIA [54], TSRLCMM [55], APO [56], RLQFAO [24], and EDBO [18]. Their configurations are also summarized in Table SI. Among these methods, TLDE, CJADE, GosDE, and ISHACDE are distinct variants of the DE algorithm; DGTCIA is a variant of the crested ibis algorithm; TSRLCMM and RLQFAO are a reinforcement learning-based methods that dynamically select evolutionary strategies.

TABLE I
STATISTICAL COMPARISON OF OPOPSO AND PSO VARIANTS ON THE CEC2017 BENCHMARK SUITE

Benchmark suite	Problem Property	Index	OPOPSO	SDLSO	TPLSO	EAPSO	SDPSO	CL-DMSPSO	L-ICSO	MFCPSO
CEC2017-30D	unimodal functions (f_1, f_3)	w/t/l	–	0/1/1	2/0/0	1/1/0	0/1/1	1/1/0	1/1/0	1/1/0
	simple multimodal functions (f_4 - f_{10})	w/t/l	–	3/3/1	7/0/0	6/0/1	6/0/1	5/2/0	5/1/1	6/1/0
	hybrid functions (f_{11} - f_{20})	w/t/l	–	2/7/1	7/2/1	6/2/2	4/3/3	7/1/2	6/3/1	8/0/2
	composition functions (f_{21} - f_{30})	w/t/l	–	3/7/0	8/1/1	8/0/2	6/3/1	7/2/1	6/4/0	5/2/3
	Overall	w/t/l	–	8/18/3	24/3/2	21/3/5	16/7/6	20/6/3	18/9/2	20/4/5
	Overall	rank	**2.60**	**2.60**	5.96	6.29	4.08	5.96	4.021	4.48
CEC2017-50D	unimodal functions (f_1, f_3)	w/t/l	–	0/1/1	2/0/0	1/0/1	0/0/2	1/1/0	1/1/0	1/1/0
	simple multimodal functions (f_4 - f_{10})	w/t/l	–	4/3/0	7/0/0	6/0/1	6/0/1	7/0/0	6/1/0	6/0/1
	hybrid functions (f_{11} - f_{20})	w/t/l	–	2/5/3	6/3/1	5/2/3	5/0/5	8/2/0	7/1/2	8/1/1
	composition functions (f_{21} - f_{30})	w/t/l	–	4/6/0	10/0/0	9/1/0	8/1/1	8/2/0	9/1/0	6/1/3
	Overall	w/t/l	–	10/15/4	25/3/1	21/3/5	19/1/9	24/5/0	23/4/2	21/3/5
	Overall	rank	**2.29**	2.83	5.92	6.25	4.52	6.00	4.08	4.10
CEC2017-100D	unimodal functions (f_1, f_3)	w/t/l	–	1/1/0	2/0/0	1/1/0	0/0/2	1/1/0	1/1/0	1/1/0
	simple multimodal functions (f_4 - f_{10})	w/t/l	–	5/2/0	7/0/0	6/0/1	6/0/1	7/0/0	7/0/0	7/0/0
	hybrid functions (f_{11} - f_{20})	w/t/l	–	4/3/3	9/1/0	6/2/2	4/3/3	8/2/0	7/3/0	7/3/0
	composition functions (f_{21} - f_{30})	w/t/l	–	7/3/0	9/1/0	7/2/1	8/1/1	8/2/0	9/1/0	6/1/3
	Overall	w/t/l	–	17/9/3	27/2/0	20/5/4	18/4/7	24/5/0	24/5/0	21/5/3
	Overall	rank	**2.00**	2.54	6.00	6.21	4.42	5.96	4.50	4.38

TABLE II
STATISTICAL COMPARISON OF OPOPSO AND OTHER ALGORITHM VARIANTS ON THE CEC2017 BENCHMARK SUITE

Benchmark suite	Problem Property	Index	OPOPSO	TLDE	CJADE	GosDE	ISHACDE	DGTCIA	TSRLCMM	APO	RLQFAO	EDBO
CEC2017-30D	unimodal functions (f_1, f_3)	w/t/l	–	0/0/2	0/1/1	1/0/1	1/0/1	1/0/1	1/1/0	1/0/1	2/0/0	2/0/0
	simple multimodal functions (f_4 - f_{10})	w/t/l	–	5/1/1	4/2/1	4/2/1	6/0/1	6/1/0	6/1/0	7/0/0	7/0/0	6/1/0
	hybrid functions (f_{11} - f_{20})	w/t/l	–	2/2/6	3/1/6	3/0/7	3/1/6	7/1/2	6/2/2	2/1/7	7/2/1	7/2/1
	composition functions (f_{21} - f_{30})	w/t/l	–	6/2/2	5/3/2	4/3/3	9/0/1	9/1/0	10/0/0	6/2/2	8/2/0	7/1/2
	Overall	w/t/l	–	13/5/11	12/7/10	12/5/12	19/1/9	23/3/3	23/4/2	16/3/10	24/4/1	22/4/3
	Overall	rank	3.53	**3.26**	3.84	3.60	6.93	9.07	5.33	3.36	7.72	8.35
CEC2017-50D	unimodal functions (f_1, f_3)	w/t/l	–	0/0/2	0/0/2	1/1/0	1/0/1	0/1/1	1/0/1	1/1/0	2/0/0	2/0/0
	simple multimodal functions (f_4 - f_{10})	w/t/l	–	5/0/2	5/2/0	6/1/0	6/1/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0
	hybrid functions (f_{11}-f_{20})	w/t/l	–	4/0/6	4/1/5	4/1/5	6/1/3	7/1/2	6/2/2	6/0/4	8/2/0	10/0/0
	composition functions (f_{21} - f_{30})	w/t/l	–	8/1/1	9/0/1	8/0/2	9/0/1	10/0/0	10/0/0	9/0/1	9/0/1	6/1/3
	Overall	w/t/l	–	17/1/11	18/3/8	19/3/7	22/2/5	24/2/3	24/2/3	23/1/5	26/2/1	24/1/3
	Overall	rank	**2.83**	3.31	3.38	4.28	5.72	7.03	8.62	4.31	7.34	8.17
CEC2017-100D	unimodal functions (f_1, f_3)	w/t/l	–	0/0/2	0/0/2	1/1/0	1/0/1	0/0/2	0/1/1	1/1/0	1/1/0	2/0/0
	simple multimodal functions (f_4 - f_{10})	w/t/l	–	6/0/1	5/1/1	6/1/0	6/1/0	7/0/0	7/0/0	7/0/0	7/0/0	7/0/0
	hybrid functions (f_{11} - f_{20})	w/t/l	–	4/2/4	4/2/4	6/2/2	9/1/0	8/0/2	8/1/1	7/3/0	9/1/0	10/0/0
	composition functions (f_{21} - f_{30})	w/t/l	–	9/0/1	7/2/1	7/2/1	9/0/1	10/0/0	10/0/0	9/0/1	10/0/0	8/2/0
	Overall	w/t/l	–	19/2/8	16/5/8	20/6/3	25/2/2	25/0/4	25/2/2	24/4/1	26/2/1	24/1/3
	Overall	rank	**2.28**	3.24	3.03	4.93	6.00	6.79	8.03	4.83	7.07	8.79

A. Comparison with PSO Variants on Benchmark Suite

In the experiment, 29 benchmark problems from CEC2017 are selected for evaluation. Following the CEC2017 evaluation criteria, the search space is defined as $[-100, 100]^D$, and $FES_{\max} = D \times 10^4$. Due to its removal from the official code, function f_2 is excluded from the subsequent experiments.

In this subsection, we compare OPOPSO with seven PSO variants on the CEC2017 benchmark suite. The average value and standard deviation of each function, run independently 35 times, are documented in the supplementary material. Additionally, the Wilcoxon rank-sum test, corrected with the Benjamini–Hochberg (BH) method, is adopted to determine statistical significance.

The test results are summarized using the symbols "+," "=," and "-," which respectively indicate that OPOPSO achieves significantly superior, comparable, or significantly inferior performance relative to the compared algorithms at a significance level of 0.05. The notation "w/t/l" denotes the total counts of these symbols. The Friedman test is conducted to compare the average ranks, in order to evaluate the overall performance of OPOPSO and its competitors across the entire benchmark suite. For clarity, Table I summarizes the results across different problem types.

As presented in Table I, the following observations can be made on the CEC2017 benchmark suite.

1) The results of the Friedman test demonstrate that OPOPSO achieves the lowest average ranking across all dimensions, demonstrating its significant superiority over the other seven PSO variants in solving the CEC2017 benchmark problems.

2) Comparison results across different problem dimensions reveal that OPOPSO achieves significantly better performance than the other PSO variants across various types of test functions. More notably, on simple multimodal functions and composition functions, OPOPSO demonstrates a clear and significant superiority over the other PSO variants.

In conclusion, tests conducted on the CEC2017 benchmark functions verify that the proposed OPOPSO algorithm achieves better performance than other existing PSO variants.

B. Comparison with Other Algorithm Variants on Benchmark Suite

This subsection compares the proposed algorithm with nine other mature variants on the CEC2017 benchmark suite. These include DE variants, evolutionary algorithms improved by reinforcement learning, and other metaheuristics algorithms. As presented in Table II, the following observations can be observed for the CEC2017 benchmark suite.

1) As indicated by the Friedman test, OPOPSO attains the lowest overall rank among the other compared algorithm variants, except in the 30-dimensional case. Furthermore, its rank value decreases progressively as the problem dimension increases. These findings confirm that OPOPSO exhibits better adaptability to higher-dimensional problems than the other algorithm variants.

2) The comparison results across different dimensions show that OPOPSO achieves markedly better performance than the seven other algorithms on simple multimodal functions and composition functions. The overall results of the Wilcoxon rank-sum test confirm the superior performance of OPOPSO.

The experimental results collectively indicate that OPOPSO achieves more favorable performance than the seven other algorithms and their variants.

C. Comparison with PSO Variants on Path Planning Problems

To address the cooperative path planning problem for UAVs, this paper divides the constraint costs in planning into two parts: single-UAV constraints and multi-UAV separation maintenance constraints. Single-UAV constraints include terrain avoidance, trajectory length, threat-zone evasion, and the UAV's own physical limits, while multi-UAV separation maintenance constraints consider the safe separation distance between UAVs.

To address the poor quality of initial populations and the low convergence speed in multi-UAV cooperative path planning, this paper presents an initialization strategy based on single-UAV pre-optimization. The core idea of the proposed scheme is as follows. First, before introducing separation maintenance constraints, each UAV conducts path pre-optimization under its own constraints. The optimization algorithm is then used to generate a set of feasible trajectories that minimize the single-UAV constraint cost. Next, the initial multi-UAV population is constructed by randomly selecting and combining these pre-optimized individual trajectories. Finally, separation maintenance constraints are incorporated to perform multi-UAV cooperative optimization, which further reduces the total trajectory cost and guarantees cooperative safety.

Eight test scenarios are constructed based on the DEM map, each involving cooperative optimization of two, three, and four UAVs. The visualizations of the specific scenarios are provided in the supplementary material. In the multi-UAV path planning experiments, the trajectory of each UAV is defined by 12 waypoints, with 10 sampling points uniformly placed between adjacent waypoints for collision detection and constraint calculation. The safety radius is set as $r_s = 8$, and the distance constraint threshold between waypoints is $r_d = 160$. The $FES_{\max}$ for single-UAV pre-optimization in each algorithm is set to 10,000 multiplied by the number of UAVs, while the $FES_{\max}$ for multi-UAV cooperative optimization is set to 40,000 multiplied by the number of UAVs. Each scenario was run independently 30 times. In addition to the mean ranks from the Friedman test, we also report the median ranks to mitigate the impact of outlier results. As shown in Table IV, the specific results of the path planning simulation experiments reveal the following findings.

1) As indicated by the Friedman test, OPOPSO achieves the lowest overall rank among all compared PSO variants and performs significantly better than the others. This indicates the effectiveness of the OPOPSO algorithm for path planning in diverse environments. Furthermore, OPOPSO maintains a low standard deviation, which indicates superior repeatability relative to other PSO variants.

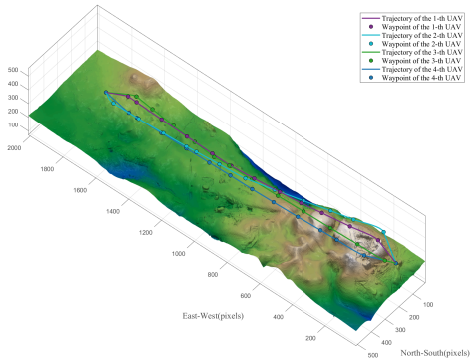
2) Regarding the overall results of the Wilcoxon rank-sum test, OPOPSO demonstrates consistent superiority. It is noteworthy that the few cases where OPOPSO is outperformed by SDLSO occur only in scenarios with 2 or 3 UAVs. In all other cases, OPOPSO demonstrates a remarkable superiority over the comparison algorithms.

TABLE III
COMPARATIVE EXPERIMENTAL RESULTS OF OPOPSO AND PSO VARIANTS UNDER VARIOUS SCENARIOS

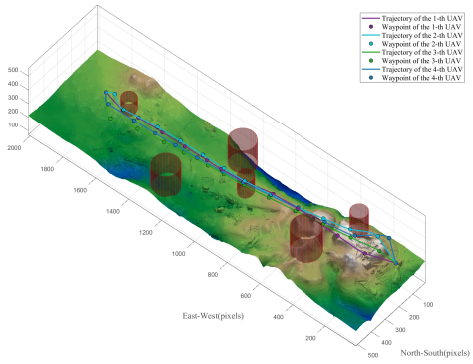
Experimental Scenario	Number of UAVs	Index	OPOPSO	SDLSO	TPLSO	EAPSO	SDPSO	CL-DMSPSO	L-ICSO	MFCPSO
Scenario 1	2	Mean	3.55e+03	**3.53e+03(-)**	3.69e+03(+)	1.39e+04(+)	5.03e+03(+)	1.52e+04(+)	3.71e+03(+)	3.59e+03(+)
		Std	7.02e+00	8.06e+00	2.44e+02	1.80e+04	1.76e+03	7.00e+03	8.31e+01	8.05e+00
		Median	3.55e+03	3.52e+03	3.61e+03	6.94e+03	4.52e+03	1.39e+04	3.69e+03	3.59e+03
	3	Mean	5.36e+03	**5.33e+03(-)**	5.75e+03(+)	2.50e+04(+)	6.64e+03(+)	2.49e+05(+)	6.33e+03(+)	5.47e+03(+)
		Std	2.08e+01	3.92e+01	4.42e+02	2.95e+04	1.86e+03	1.21e+06	3.37e+02	3.91e+01
		Median	5.36e+03	5.32e+03	5.59e+03	1.25e+04	6.03e+03	2.55e+04	6.28e+03	5.48e+03
	4	Mean	**7.24e+03**	7.33e+03(=)	7.77e+03(+)	4.96e+04(+)	9.42e+03(+)	3.86e+04(+)	2.02e+04(+)	7.43e+03(+)
		Std	7.42e+01	2.13e+02	1.06e+03	6.22e+04	2.47e+03	1.48e+04	7.56e+03	9.34e+01
		Median	7.22e+03	7.26e+03	7.53e+03	2.04e+04	8.42e+03	3.85e+04	1.80e+04	7.39e+03
Scenario 2	2	Mean	**3.60e+03**	2.02e+06(=)	8.45e+06(+)	6.47e+04(+)	4.97e+03(+)	1.87e+07(+)	6.62e+06(+)	3.64e+03(+)
		Std	7.90e+00	1.10e+07	2.19e+07	2.62e+05	2.65e+03	1.77e+07	2.01e+07	1.63e+01
		Median	3.60e+03	3.58e+03	2.45e+04	7.12e+03	4.00e+03	1.88e+07	1.88e+04	3.63e+03
	3	Mean	**5.49e+03**	6.34e+06(+)	6.08e+06(+)	5.29e+06(+)	7.92e+03(+)	4.50e+07(+)	3.32e+06(+)	5.62e+03(+)
		Std	6.35e+01	2.41e+07	2.30e+07	2.88e+07	1.80e+03	4.28e+07	1.81e+07	5.35e+01
		Median	5.48e+03	5.66e+03	2.43e+04	1.76e+04	7.46e+03	5.55e+07	1.37e+04	5.61e+03
	4	Mean	**7.53e+03**	9.38e+03(+)	1.30e+07(+)	4.94e+04(+)	1.04e+04(+)	7.49e+07(+)	8.46e+06(+)	7.68e+03(+)
		Std	1.27e+02	3.56e+03	3.95e+07	6.27e+04	2.73e+03	9.36e+07	3.21e+07	1.31e+02
		Median	7.51e+03	7.99e+03	4.35e+04	2.87e+04	9.68e+03	7.41e+07	1.68e+04	7.63e+03
Scenario 3	2	Mean	**3.60e+03**	4.23e+03(-)	2.35e+04(+)	2.14e+04(+)	5.62e+03(+)	2.96e+07(+)	2.03e+06(+)	3.64e+03(+)
		Std	6.02e+00	1.37e+03	1.47e+04	3.73e+04	3.63e+03	1.66e+07	1.10e+07	1.70e+01
		Median	3.60e+03	3.59e+03	2.05e+04	8.49e+03	4.30e+03	3.70e+07	1.72e+04	3.63e+03
	3	Mean	**5.50e+03**	6.35e+03(+)	2.93e+04(+)	4.31e+04(+)	7.25e+03(+)	6.36e+07(+)	3.04e+06(+)	5.62e+03(+)
		Std	5.60e+01	2.14e+03	1.99e+04	6.72e+04	2.62e+03	7.40e+07	1.66e+07	5.89e+01
		Median	5.49e+03	5.65e+03	2.44e+04	2.03e+04	6.07e+03	5.56e+07	1.32e+04	5.61e+03
	4	Mean	**7.50e+03**	8.89e+03(+)	4.70e+04(+)	5.26e+05(+)	9.73e+03(+)	1.03e+08(+)	4.06e+06(+)	7.80e+03(+)
		Std	1.37e+02	3.00e+03	3.57e+04	2.49e+06	2.19e+03	9.75e+07	2.21e+07	1.35e+02
		Median	7.50e+03	8.01e+03	4.12e+04	3.75e+04	9.03e+03	7.43e+07	1.55e+04	7.80e+03
Scenario 4	2	Mean	**3.68e+03**	4.06e+06(+)	1.89e+07(+)	3.14e+04(+)	7.33e+03(+)	3.64e+08(+)	1.28e+07(+)	3.74e+03(+)
		Std	3.17e+01	1.54e+07	2.94e+07	7.11e+04	3.54e+03	1.83e+08	2.58e+07	3.19e+01
		Median	3.67e+03	2.37e+04	5.80e+04	7.74e+03	5.80e+03	3.31e+08	8.76e+04	3.73e+03
	3	Mean	**5.64e+03**	3.05e+06(+)	2.81e+07(+)	1.80e+08(+)	1.09e+04(+)	6.28e+08(+)	2.60e+07(+)	5.76e+03(+)
		Std	1.18e+02	1.66e+07	4.37e+07	9.83e+08	5.49e+03	4.86e+08	4.37e+07	1.00e+02
		Median	5.59e+03	1.91e+04	9.78e+04	1.54e+04	9.34e+03	4.97e+08	8.70e+04	5.75e+03
	4	Mean	**7.77e+03**	7.42e+06(+)	2.47e+07(+)	4.06e+12(+)	1.46e+04(+)	6.74e+08(+)	4.00e+07(+)	8.18e+03(+)
		Std	1.96e+02	4.04e+07	5.01e+07	2.22e+13	6.03e+03	4.12e+08	6.19e+07	1.80e+02
		Median	7.72e+03	3.10e+04	1.04e+05	2.75e+04	1.29e+04	6.44e+08	1.62e+05	8.18e+03
Scenario 5	2	Mean	3.01e+03	**2.98e+03(-)**	3.07e+03(+)	4.80e+03(+)	3.09e+03(+)	3.45e+03(+)	3.12e+03(+)	3.06e+03(+)
		Std	1.03e+01	5.74e+00	6.96e+01	5.05e+03	3.01e+01	4.37e+02	2.16e+01	1.42e+01
		Median	3.01e+03	2.98e+03	3.03e+03	3.20e+03	3.08e+03	3.38e+03	3.12e+03	3.06e+03
	3	Mean	4.54e+03	**4.51e+03(-)**	4.71e+03(+)	1.05e+04(+)	4.69e+03(+)	5.13e+03(+)	5.57e+03(+)	4.62e+03(+)
		Std	1.46e+01	1.46e+01	1.44e+02	1.63e+04	6.36e+01	8.09e+02	4.15e+02	2.09e+01
		Median	4.54e+03	4.51e+03	4.70e+03	4.91e+03	4.67e+03	4.78e+03	5.44e+03	4.61e+03
	4	Mean	**6.11e+03**	6.13e+03(=)	6.38e+03(+)	2.50e+04(+)	6.52e+03(+)	6.64e+03(+)	2.26e+04(+)	6.25e+03(+)
		Std	2.90e+01	1.02e+02	2.40e+02	5.86e+04	1.93e+02	5.09e+02	5.54e+03	3.69e+01
		Median	6.11e+03	6.11e+03	6.30e+03	7.01e+03	6.48e+03	6.59e+03	2.16e+04	6.24e+03
Scenario 6	2	Mean	**3.09e+03**	3.15e+03(+)	6.82e+03(+)	4.54e+03(+)	3.32e+03(+)	3.36e+06(+)	6.52e+03(+)	3.14e+03(+)
		Std	1.59e+01	3.57e+01	6.45e+03	2.25e+03	2.00e+02	1.83e+07	5.90e+03	2.19e+01
		Median	3.09e+03	3.16e+03	4.39e+03	3.54e+03	3.22e+03	1.42e+04	4.74e+03	3.13e+03
	3	Mean	**4.68e+03**	4.80e+03(+)	9.46e+03(+)	1.04e+04(+)	5.14e+03(+)	1.67e+07(+)	8.28e+03(+)	4.77e+03(+)
		Std	2.49e+01	5.79e+01	8.49e+03	1.47e+04	8.32e+02	6.48e+07	7.74e+03	2.96e+01
		Median	4.68e+03	4.79e+03	6.05e+03	5.62e+03	4.86e+03	2.42e+04	6.33e+03	4.77e+03
	4	Mean	**6.36e+03**	6.58e+03(+)	8.93e+03(+)	6.31e+05(+)	6.89e+03(+)	2.34e+07(+)	1.43e+04(+)	6.50e+03(+)
		Std	6.21e+01	1.73e+02	3.45e+03	3.38e+06	2.40e+02	8.98e+07	3.49e+03	3.70e+01
		Median	6.35e+03	6.51e+03	7.97e+03	8.38e+03	6.87e+03	3.82e+04	1.28e+04	6.50e+03
Scenario 7	2	Mean	**3.10e+03**	8.23e+03(+)	7.40e+03(+)	4.91e+03(+)	3.25e+03(+)	3.00e+07(+)	1.58e+04(+)	3.14e+03(+)
		Std	2.13e+01	1.14e+04	9.97e+03	3.68e+03	1.82e+02	6.51e+07	1.53e+04	1.40e+01
		Median	3.11e+03	3.19e+03	3.60e+03	3.50e+03	3.19e+03	3.92e+04	5.49e+03	3.15e+03
	3	Mean	**4.72e+03**	1.68e+04(+)	2.58e+04(+)	1.10e+04(+)	4.94e+03(+)	9.67e+07(+)	2.50e+04(+)	4.79e+03(+)
		Std	2.20e+01	2.02e+04	3.32e+04	1.26e+04	1.37e+02	1.25e+08	2.14e+04	2.64e+01
		Median	4.72e+03	4.87e+03	8.53e+03	5.61e+03	4.91e+03	7.43e+04	7.80e+03	4.79e+03
	4	Mean	**6.44e+03**	1.47e+04(+)	1.65e+04(+)	1.97e+04(+)	6.86e+03(+)	1.37e+08(+)	4.21e+04(+)	6.52e+03(+)
		Std	7.96e+01	2.04e+04	1.93e+04	3.08e+04	1.94e+02	1.77e+08	3.08e+04	5.28e+01
		Median	6.42e+03	6.83e+03	9.45e+03	9.54e+03	6.83e+03	9.90e+04	2.14e+04	6.52e+03
Scenario 8	2	Mean	**3.11e+03**	1.35e+04(+)	2.39e+04(+)	9.77e+03(+)	3.73e+03(+)	8.01e+07(+)	2.50e+04(+)	3.19e+03(+)
		Std	2.61e+01	1.59e+04	1.74e+04	2.01e+04	7.18e+02	9.61e+07	2.82e+04	6.34e+01
		Median	3.11e+03	3.18e+03	3.49e+04	3.60e+03	3.42e+03	7.00e+04	1.19e+04	3.18e+03
	3	Mean	**4.87e+03**	1.53e+04(=)	3.65e+04(+)	1.83e+04(+)	6.44e+03(+)	1.73e+08(+)	3.44e+04(+)	4.95e+03(+)
		Std	8.12e+01	2.12e+04	2.93e+04	3.32e+04	2.24e+03	1.51e+08	2.98e+04	6.45e+01
		Median	4.89e+03	4.85e+03	3.77e+04	5.95e+03	5.40e+03	3.00e+08	2.93e+04	4.92e+03
	4	Mean	**6.62e+03**	2.65e+04(+)	4.22e+04(+)	3.98e+04(+)	7.64e+03(+)	1.60e+08(+)	5.90e+04(+)	6.96e+03(+)
		Std	1.01e+02	3.03e+04	4.06e+04	1.17e+05	9.33e+02	2.01e+08	3.92e+04	1.14e+02
		Median	6.61e+03	7.32e+03	1.77e+04	9.18e+03	7.39e+03	1.16e+05	7.24e+04	6.97e+03
		w/t/l	-	15/4/5	24/0/0	24/0/0	24/0/0	24/0/0	24/0/0	24/0/0
		rank	**1.17**	3.42	5.54	5.96	3.83	7.71	6.13	2.25
		median ranks	**1.29**	2.63	5.83	5.63	4.08	7.83	6.29	2.42

TABLE IV
COMPARATIVE EXPERIMENTAL RESULTS OF OPOPSO AND OTHER ALGORITHM VARIANTS UNDER VARIOUS SCENARIOS

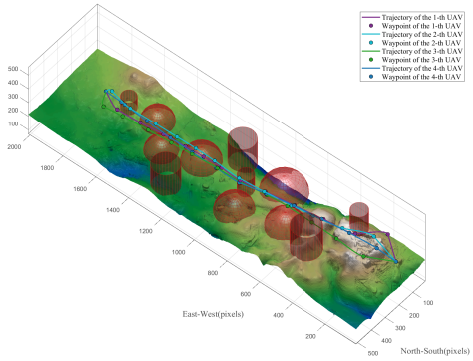
Experimental Scenario	Number of UAVs	Index	OPOPSO	TLDE	CJADE	GosDE	ISHACDE	DGTCIA	TSRLCMM	APO	RLQFAO	EDBO
Scenario 1	2	Mean	3.55e+03	**3.53e+03(-)**	3.53e+03(-)	3.56e+03(=)	3.57e+03(+)	3.66e+03(+)	1.59e+04(+)	5.03e+03(+)	1.34e+05(+)	2.41e+05(+)
		Std	7.02e+00	7.88e+00	9.93e+00	2.14e+01	2.80e+01	1.54e+02	9.22e+03	1.76e+03	1.63e+04	2.12e+05
		Median	3.55e+03	3.53e+03	3.53e+03	3.55e+03	3.56e+03	3.60e+03	1.38e+04	4.52e+03	1.33e+05	1.70e+05
	3	Mean	5.36e+03	**5.34e+03(-)**	5.37e+03(=)	5.40e+03(+)	5.71e+03(+)	7.00e+03(+)	2.63e+04(+)	6.64e+03(+)	2.69e+05(+)	7.80e+05(+)
		Std	2.08e+01	3.31e+01	7.37e+01	4.37e+01	3.13e+02	3.03e+03	1.51e+04	1.86e+03	9.64e+04	3.50e+05
		Median	5.36e+03	5.34e+03	5.34e+03	5.40e+03	5.62e+03	6.02e+03	2.36e+04	6.03e+03	2.35e+05	7.30e+05
	4	Mean	**7.24e+03**	7.25e+03(=)	7.53e+03(+)	7.29e+03(+)	8.70e+03(+)	1.03e+06(+)	3.98e+04(+)	9.42e+03(+)	1.77e+06(+)	9.44e+05(+)
		Std	7.42e+01	1.01e+02	4.22e+02	9.73e+01	9.58e+02	5.59e+06	1.59e+04	2.47e+03	4.22e+06	3.68e+05
		Median	7.22e+03	7.22e+03	7.39e+03	7.27e+03	8.30e+03	1.00e+04	4.06e+04	8.42e+03	3.71e+05	8.86e+05
Scenario 2	2	Mean	**3.60e+03**	8.06e+03(+)	3.60e+03(-)	3.62e+03(=)	4.45e+03(+)	2.13e+06(+)	8.24e+04(+)	4.97e+03(+)	1.37e+05(+)	2.21e+05(+)
		Std	7.90e+00	4.86e+03	5.33e+01	4.24e+01	1.16e+03	1.14e+07	2.51e+04	2.65e+03	1.88e+04	1.95e+05
		Median	3.60e+03	6.61e+03	3.58e+03	3.61e+03	3.92e+03	3.98e+03	8.34e+04	4.00e+03	1.34e+05	1.27e+05
	3	Mean	**5.49e+03**	1.67e+04(+)	5.56e+03(=)	5.53e+03(+)	6.40e+03(+)	3.73e+06(+)	6.23e+06(+)	7.92e+03(+)	4.06e+05(+)	9.71e+05(+)
		Std	6.35e+01	8.96e+03	1.79e+02	7.06e+01	9.13e+02	1.23e+07	2.31e+07	1.80e+03	9.26e+05	5.39e+05
		Median	5.48e+03	1.49e+04	5.49e+03	5.52e+03	6.16e+03	7.72e+03	1.31e+05	7.46e+03	2.32e+05	8.51e+05
	4	Mean	**7.53e+03**	1.59e+04(+)	8.07e+03(+)	7.66e+03(=)	9.27e+03(+)	2.95e+06(+)	1.33e+07(+)	1.04e+04(+)	3.55e+05(+)	1.24e+06(+)
		Std	1.27e+02	1.58e+04	8.66e+02	4.25e+02	1.15e+03	1.27e+07	3.98e+07	2.73e+03	8.11e+04	4.92e+05
		Median	7.51e+03	1.03e+04	7.87e+03	7.58e+03	8.87e+03	1.16e+04	1.98e+05	9.68e+03	3.23e+05	1.19e+06
Scenario 3	2	Mean	**3.60e+03**	7.71e+03(+)	3.60e+03(-)	3.62e+03(=)	4.56e+03(+)	5.67e+06(+)	1.10e+06(+)	5.62e+03(+)	1.37e+05(+)	2.37e+05(+)
		Std	6.02e+00	6.20e+03	4.24e+01	2.31e+01	1.32e+03	1.56e+07	5.57e+06	3.63e+03	1.90e+04	2.09e+05
		Median	3.60e+03	5.18e+03	3.59e+03	3.61e+03	3.90e+03	1.03e+04	8.87e+04	4.30e+03	1.33e+05	1.40e+05
	3	Mean	**5.50e+03**	1.13e+04(+)	5.52e+03(=)	5.64e+03(+)	6.73e+03(+)	1.06e+07(+)	3.32e+06(+)	7.25e+03(+)	2.70e+05(+)	9.74e+05(+)
		Std	5.60e+01	6.51e+03	1.19e+02	4.81e+02	1.29e+03	3.26e+07	1.72e+07	2.62e+03	2.14e+05	5.71e+05
		Median	5.49e+03	9.01e+03	5.50e+03	5.52e+03	6.26e+03	1.39e+04	1.49e+05	6.07e+03	2.31e+05	7.94e+05
	4	Mean	**7.50e+03**	9.95e+03(+)	8.01e+03(+)	7.73e+03(+)	9.38e+03(+)	1.35e+06(+)	2.53e+05(+)	9.73e+03(+)	3.77e+05(+)	1.29e+06(+)
		Std	1.37e+02	4.45e+03	4.31e+02	7.22e+02	8.83e+02	5.07e+06	1.27e+05	2.19e+03	1.16e+05	5.32e+05
		Median	7.50e+03	8.05e+03	7.96e+03	7.60e+03	9.34e+03	1.10e+04	2.31e+05	9.03e+03	3.35e+05	1.09e+06
Scenario 4	2	Mean	**3.68e+03**	1.10e+04(+)	3.69e+03(=)	3.77e+03(+)	1.65e+07(+)	5.70e+07(+)	1.79e+05(+)	7.33e+03(+)	1.31e+05(+)	2.04e+05(+)
		Std	3.17e+01	1.01e+04	6.13e+01	8.90e+01	9.00e+07	7.18e+07	2.23e+05	3.54e+03	1.10e+04	1.06e+05
		Median	3.67e+03	9.43e+03	3.67e+03	3.75e+03	5.02e+04	1.30e+07	1.00e+05	5.80e+03	1.29e+05	1.52e+05
	3	Mean	**5.64e+03**	1.27e+04(+)	5.68e+03(=)	6.46e+03(+)	2.36e+07(+)	2.00e+08(+)	3.18e+06(+)	1.09e+04(+)	2.29e+05(+)	1.01e+06(+)
		Std	1.18e+02	4.37e+03	1.61e+02	1.95e+03	1.29e+08	2.50e+08	1.66e+07	5.49e+03	2.37e+04	4.47e+05
		Median	5.59e+03	1.28e+04	5.64e+03	5.75e+03	7.69e+04	9.90e+07	1.46e+05	9.34e+03	2.28e+05	8.82e+05
	4	Mean	**7.77e+03**	1.55e+04(+)	8.11e+03(+)	8.82e+03(=)	2.68e+07(+)	3.99e+08(+)	4.25e+06(+)	1.46e+04(+)	4.12e+05(+)	1.20e+06(+)
		Std	1.96e+02	2.63e+04	3.60e+02	2.29e+03	1.29e+08	6.79e+08	2.21e+07	6.03e+03	3.31e+05	4.91e+05
		Median	7.72e+03	9.83e+03	8.07e+03	7.90e+03	1.05e+05	3.26e+04	2.26e+05	1.29e+04	3.27e+05	1.04e+06
Scenario 5	2	Mean	3.01e+03	2.99e+03(-)	**2.98e+03(-)**	3.06e+03(+)	3.02e+03(+)	3.02e+03(+)	8.36e+03(+)	3.09e+03(+)	8.01e+04(+)	8.21e+03(+)
		Std	1.03e+01	8.56e+00	7.25e+00	3.17e+01	1.28e+01	1.61e+01	3.54e+03	3.01e+01	5.70e+03	4.55e+03
		Median	3.01e+03	2.99e+03	2.98e+03	3.05e+03	3.02e+03	3.02e+03	8.65e+03	3.08e+03	8.09e+04	7.32e+03
	3	Mean	4.54e+03	**4.53e+03(-)**	4.59e+03(+)	4.60e+03(+)	4.64e+03(+)	4.63e+03(+)	2.20e+04(+)	4.69e+03(+)	1.35e+05(+)	5.64e+04(+)
		Std	1.46e+01	1.29e+01	6.06e+01	2.16e+01	1.10e+02	4.11e+01	1.01e+04	6.36e+01	8.40e+03	4.42e+04
		Median	4.54e+03	4.53e+03	4.59e+03	4.60e+03	4.60e+03	4.62e+03	1.93e+04	4.67e+03	1.36e+05	3.66e+04
	4	Mean	**6.11e+03**	6.10e+03(=)	6.40e+03(+)	6.17e+03(+)	7.14e+03(+)	6.78e+03(+)	2.40e+04(+)	6.52e+03(+)	2.01e+05(+)	1.24e+05(+)
		Std	2.90e+01	2.64e+01	3.95e+02	4.11e+01	6.23e+02	7.28e+02	1.48e+04	1.93e+02	1.38e+04	9.92e+04
		Median	6.11e+03	6.10e+03	6.24e+03	6.17e+03	6.95e+03	6.44e+03	2.06e+04	6.48e+03	2.00e+05	9.15e+04
Scenario 6	2	Mean	3.09e+03	3.32e+03(+)	**3.07e+03(-)**	3.14e+03(+)	3.27e+03(+)	2.20e+04(+)	4.32e+04(+)	3.32e+03(+)	8.15e+04(+)	1.20e+04(+)
		Std	1.59e+01	2.09e+02	2.89e+01	3.14e+01	2.28e+02	1.84e+04	1.78e+04	2.00e+02	4.53e+03	6.58e+03
		Median	3.09e+03	3.20e+03	3.06e+03	3.13e+03	3.20e+03	3.15e+04	3.96e+04	3.22e+03	8.11e+04	1.07e+04
	3	Mean	**4.68e+03**	4.92e+03(+)	4.72e+03(+)	4.78e+03(+)	5.17e+03(+)	2.99e+04(+)	8.64e+04(+)	5.14e+03(+)	1.38e+05(+)	1.15e+05(+)
		Std	2.49e+01	1.64e+02	7.07e+01	6.30e+01	3.19e+02	2.72e+04	3.37e+04	8.32e+02	7.71e+03	9.07e+04
		Median	4.68e+03	4.86e+03	4.71e+03	4.77e+03	5.12e+03	3.01e+04	8.94e+04	4.86e+03	1.36e+05	9.18e+04
	4	Mean	**6.36e+03**	6.54e+03(+)	6.77e+03(+)	6.47e+03(+)	8.09e+03(+)	3.36e+04(+)	1.11e+05(+)	6.89e+03(+)	1.98e+05(+)	2.21e+05(+)
		Std	6.21e+01	1.64e+02	3.86e+02	1.25e+02	7.34e+02	3.31e+04	5.42e+04	2.40e+02	1.02e+04	1.23e+05
		Median	6.35e+03	6.51e+03	6.69e+03	6.43e+03	7.93e+03	8.52e+03	1.12e+05	6.87e+03	1.98e+05	2.28e+05
Scenario 7	2	Mean	3.10e+03	3.31e+03(+)	**3.08e+03(-)**	3.14e+03(+)	6.29e+03(+)	1.89e+04(+)	5.10e+04(+)	3.25e+03(+)	8.03e+04(+)	1.08e+04(+)
		Std	2.13e+01	2.50e+02	2.85e+01	2.67e+01	8.40e+03	2.15e+04	1.49e+04	1.82e+02	4.92e+03	7.01e+03
		Median	3.11e+03	3.22e+03	3.08e+03	3.14e+03	3.29e+03	3.66e+03	5.17e+04	3.19e+03	8.04e+04	7.81e+03
	3	Mean	**4.72e+03**	4.91e+03(+)	4.75e+03(=)	4.90e+03(+)	9.53e+03(+)	2.70e+04(+)	8.60e+04(+)	4.94e+03(+)	1.39e+05(+)	1.06e+05(+)
		Std	2.20e+01	8.76e+01	8.19e+01	4.39e+02	1.28e+04	2.40e+04	3.07e+04	1.37e+02	8.28e+03	7.40e+04
		Median	4.72e+03	4.89e+03	4.75e+03	4.77e+03	5.32e+03	6.40e+03	9.32e+04	4.91e+03	1.38e+05	9.12e+04
	4	Mean	**6.44e+03**	6.67e+03(+)	6.86e+03(+)	6.66e+03(+)	1.40e+04(+)	3.68e+04(+)	1.20e+05(+)	6.86e+03(+)	1.97e+05(+)	2.30e+05(+)
		Std	7.96e+01	2.20e+02	3.09e+02	6.65e+02	1.73e+04	3.03e+04	2.94e+04	1.94e+02	1.36e+04	1.14e+05
		Median	6.42e+03	6.61e+03	6.85e+03	6.47e+03	8.22e+03	1.68e+04	1.25e+05	6.83e+03	1.97e+05	2.07e+05
Scenario 8	2	Mean	**3.11e+03**	3.45e+03(+)	3.16e+03(=)	3.30e+03(+)	1.61e+04(+)	6.69e+06(+)	6.83e+04(+)	3.73e+03(+)	8.41e+04(+)	1.97e+04(+)
		Std	2.61e+01	7.90e+02	1.11e+02	4.47e+02	2.66e+04	2.54e+07	4.47e+04	7.18e+02	4.65e+03	1.14e+04
		Median	3.11e+03	3.20e+03	3.12e+03	3.23e+03	3.28e+03	1.20e+04	6.24e+04	3.42e+03	8.41e+04	1.77e+04
	3	Mean	**4.87e+03**	4.91e+03(=)	4.93e+03(=)	5.63e+03(=)	2.02e+04(+)	2.34e+07(+)	9.57e+04(+)	6.44e+03(+)	1.44e+05(+)	1.58e+05(+)
		Std	8.12e+01	1.36e+02	1.31e+02	1.64e+03	3.18e+04	6.26e+07	2.54e+04	2.24e+03	9.48e+03	1.07e+05
		Median	4.89e+03	4.85e+03	4.92e+03	4.92e+03	5.60e+03	4.87e+04	9.87e+04	5.40e+03	1.46e+05	1.41e+05
	4	Mean	**6.62e+03**	6.65e+03(=)	7.15e+03(+)	7.68e+03(+)	2.55e+04(+)	1.34e+07(+)	1.43e+05(+)	7.64e+03(+)	2.03e+05(+)	3.25e+05(+)
		Std	1.01e+02	1.28e+02	3.49e+02	1.87e+03	3.07e+04	3.46e+07	3.72e+04	9.33e+02	1.38e+04	1.78e+05
		Median	6.61e+03	6.60e+03	7.08e+03	6.78e+03	9.81e+03	1.01e+04	1.56e+05	7.39e+03	2.03e+05	2.95e+05
		w/t/l	-	16/4/4	10/8/6	18/6/0	24/0/0	24/0/0	24/0/0	24/0/0	24/0/0	24/0/0
		rank	**1.46**	3.79	2.46	3.17	5.63	8.33	8.13	5.13	8.46	8.46
		median rank	**1.67**	3.58	2.38	3.17	5.33	6.83	8.13	5.33	9.33	9.25



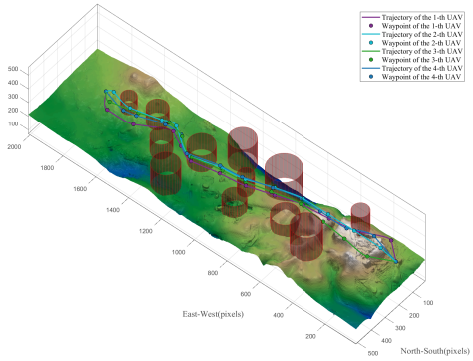
(a) Scenario 1



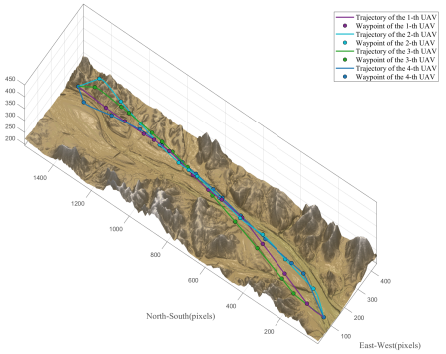
(b) Scenario 2



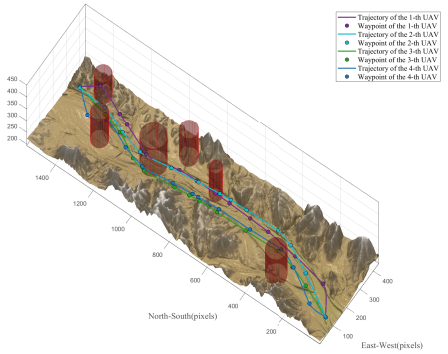
(c) Scenario 3



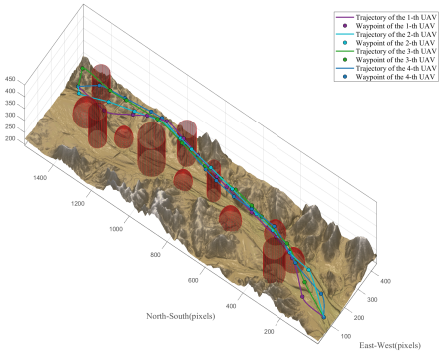
(d) Scenario 4



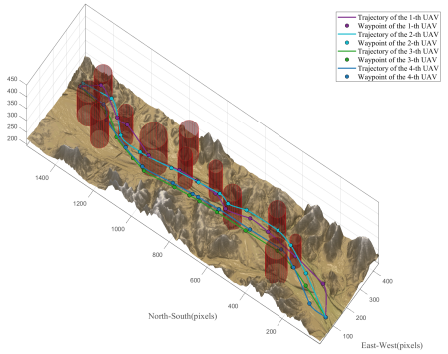
(e) Scenario 5



(f) Scenario 6



(g) Scenario 7



(h) Scenario 8

Fig. 3. Visualization of Path Planning for Four UAVs Based on the OPOPSO Algorithm

D. Comparison with Other Algorithm Variants on Path Planning Problems

To systematically assess the overall performance of the proposed OPOPSO algorithm, comparative path planning simulation experiments are performed against several other optimization algorithms. The compared algorithms include TLDE, CJADE, GosDE, ISHACDE, DGTCIA, TSRLCMM, and APO. All compared algorithms are executed on the same experimental platform, using identical evaluation metrics to ensure the fairness and validity of the comparison. The specific results are presented in Table IV.

1) Based on the outcomes of the Friedman test, OPOPSO attains the lowest overall rank among all compared algorithm variants. Notably, although the Friedman test shows that TLDE and APO outperformed OPOPSO on the 30-dimensional CEC2017 benchmark suite, their ranks for the path planning problem are 3.79 and 5.33, respectively. In contrast, OPOPSO achieves a rank of 1.46, which is markedly superior to these competitors.

2) Regarding the overall outcomes of the Wilcoxon rank-sum test, OPOPSO surpasses all other algorithms in all the path planning simulation experiments, except for TLDE, CJADE, and GosDE. In the scenario involving four UAVs, OPOPSO demonstrates significant superiority over all compared algorithms. These statistical results confirm the superior performance of OPOPSO.

The large mean values observed for some competing algorithms in Tables III and IV arise from the soft penalty for obstacle clearance, which is inversely proportional to the squared distance from a path point to the nearest obstacle or other UAVs. Consequently, algorithms that produce near-obstacle paths incur disproportionately large penalties. To mitigate the influence of such outliers, we supplement the means with medians in all statistical tables. The median-based comparisons, being robust to extreme penalties, corroborate our findings and confirm the effectiveness of OPOPSO.

E. Ablation Study

In this section, the contributions of the orthogonal perturbation operator and the dual reward mechanism to the performance of OPOPSO are examined through controlled experiments.

To validate the effectiveness of the proposed dual reward mechanism, we develop several OPOPSO variants. The first variant, denoted as OPOPSO$_1$, adopts Eq. (34) as the reward function, which is analogous to an internal reward and focuses on the fitness improvement of individual particles. The efficacy of this method has been demonstrated in [21] and [22]. We also design OPOPSO$_2$, which adopts Eq. (35) as its reward function, and this strategy has been proven effective in [57].

$$r_i = \begin{cases} 1, & f(X_i^k) > f(X_i^{k+1}) \\ -1, & \text{others} \end{cases} \tag{33}$$

$$r = \sum_i r_i \tag{34}$$

$$r = \frac{f(\hat{X}^k) - f(\hat{X}^{k+1})}{f(\hat{X}^k)}. \tag{35}$$

Furthermore, since particles in OPOPSO are influenced by the orthogonal perturbation operator, we design OPOPSO$_3$ without this operator to examine its contribution. Finally, we design OPOPSO$_4$ without reinforcement learning training, which randomly selects the parameters $\theta_{\max}$, the topology size and ϕ in each iteration. Notably, OPOPSO$_1$, OPOPSO$_2$ and OPOPSO$_3$ all use retrained neural networks for parameter selection, with their entire training process kept identical to that of the original OPOPSO.

Using the above variants, we conduct path planning simulation experiments to compare OPOPSO with its four developed versions. Table SVIII presents the comparison results, which reveal the following.

1) In terms of both the overall average rank and the statistical outcomes of the Wilcoxon rank-sum test, OPOPSO consistently achieves the most favorable results. This substantiates its significant effectiveness and superior performance across the path planning simulation experiments.

2) Comparative analyses between OPOPSO and its variants (OPOPSO$_1$ and OPOPSO$_2$) reveal a clear performance advantage for the former. This verifies the efficacy of the dual reward mechanism integrated into OPOPSO, which combines internal and external rewards to provide more comprehensive guidance for the search process.

3) The comparison between OPOPSO and OPOPSO$_3$ demonstrates the distinct superiority of the version employing the orthogonal perturbation operator. The performance gap validates the critical role of this operator in enhancing search diversity and escaping local optima. Finally, the significant performance difference observed between OPOPSO and OPOPSO$_4$ underscores both the feasibility and the importance of utilizing a reinforcement learning mechanism for adaptive parameter selection within the algorithm.

F. Sensitivity Analysis

As the sole parameter, population size is evaluated for its impact on OPOPSO performance via comparisons with variants of varying population sizes.The comparative results are presented in Table SIX. According to the statistical outcomes of the Wilcoxon rank-sum test, OPOPSO shows no significant difference from its variants with similar population sizes. However, based on the Friedman test results, a population size of 1000 yields better performance. The experimental findings suggest that a population size of 1000 is a well-suited configuration. Therefore, this study recommends setting the population size to 1000.

In addition to population size, the configuration of the D3QN action space can affect algorithm performance. We therefore conducted a sensitivity analysis by varying the upper bounds of two key parameters relative to the baseline setting ($\phi_{max} = 0.4$ and $\theta_{\max} = 0.75$). Specifically, we generated six variants: three with $\phi_{\max}$ set to 0.2, 0.6, and 0.8 while $\theta_{\max}$ remained fixed at 0.75, and three with $\theta_{\max}$ set to 1.5, 1.125, and 0.375 while $\phi_{\max}$ remained fixed at 0.4. To ensure a fair comparison, we kept the number of discrete actions identical across all D3QN agents by adjusting the step size according to the upper bound of each configuration. All agents were retrained under identical conditions.

Overall, the default parameter setting achieves a favorable exploration–exploitation trade-off. As shown in Table SX, the baseline configuration generally yields the most competitive average performance across most scenarios. Nevertheless, the variations in solution quality under different $\phi_{\max}$ and $\theta_{\max}$ settings are remarkably small, with most deviations falling well within the standard deviation ranges of repeated runs.

Notably, performance does not show a consistent worsening trend as the bounds increase or decrease. This indicates that OPOPSO is robust to the specific choices of these two parameters. Given its superior average performance and the robustness confirmed above, we recommend the baseline configuration ($\phi_{\max} = 0.4$, $\theta_{\max} = 0.75$) for practical use.

VI. Conclusion

This paper proposes an OPOPSO algorithm integrated with deep reinforcement learning, to address the challenge of multi-UAV 3D path planning in complex environments. To address the limitations of traditional PSO, OPOPSO incorporates two core enhancements. The first is an orthogonal perturbation operator, which breaks rigid update patterns and helps avoid local optima. The second is a dynamic topology reorganization mechanism, which enhances population diversity through random neighborhood construction and diverse exemplar selection. Additionally, a D3QN adaptive parameter control framework with a dual reward mechanism is designed to solve the sparse reward problem, enabling stable learning of effective parameter adjustment strategies.

To validate the superiority of OPOPSO, we evaluate the algorithm on the CEC2017 benchmark suite and path planning simulations. The simulations are constructed using a 5-meter resolution grid DEM generated from Australian LiDAR data. Against 16 advanced algorithms, including PSO variants, DE variants, and other metaheuristics, OPOPSO exhibits superior solution quality and robustness. Ablation studies confirm the effectiveness of key components: the orthogonal perturbation operator enhances search diversity and the dual reward mechanism strengthens the stability of reinforcement learning-based parameter control. Moreover, OPOPSO shows good adaptability to different problem scales and complex 3D environments, maintaining competitive performance with increasing numbers of UAVs and rising environmental complexity.

Future research will concentrate on extending OPOPSO to large-scale multi-UAV cooperative missions, where both path length and carbon-footprint efficiency ratio are considered as optimization objectives under more complex constraints such as dynamic obstacles and energy limitations. Additionally, integrating OPOPSO with advanced intelligent algorithms such as graph neural networks for environmental modeling provides a promising way to further enhance its performance in complex path planning scenarios.

References

[1] M. Morin, I. Abi-Zeid, and C.-G. Quimper, "Ant colony optimization for path planning in search and rescue operations," *European Journal of Operational Research*, vol. 305, no. 1, pp. 53–63, 2023.

[2] X. Zhang, Y. Fang, X. Zhang, J. Jiang, and X. Chen, "A novel geometric hierarchical approach for dynamic visual servoing of quadrotors," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 5, pp. 3840–3849, 2020.

[3] F. Wang, H. Li, and H. Xiong, "Truck–drone routing problem with stochastic demand," *European Journal of Operational Research*, vol. 322, no. 3, pp. 854–869, 2025.

[4] C. Huang, X. Zhou, X. Ran, J. Wang, H. Chen, and W. Deng, "Adaptive cylinder vector particle swarm optimization with differential evolution for uav path planning," *Engineering Applications of Artificial Intelligence*, vol. 121, p. 105942, 2023.

[5] C. Xu, M. Xu, and C. Yin, "Optimized multi-uav cooperative path planning under the complex confrontation environment," *Computer Communications*, vol. 162, pp. 196–203, 2020.

[6] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[7] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings*, vol. 2, 2000, pp. 995–1001.

[8] Y. Jiang, X. X. Xu, M. Y. Zheng *et al.*, "Evolutionary computation for unmanned aerial vehicle path planning: a survey," *Artificial Intelligence Review*, vol. 57, p. 267, 2024.

[9] Z.-H. Zhan *et al.*, "Matrix-based evolutionary computation," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 2, pp. 315–328, 2022.

[10] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[11] W. Du *et al.*, "Network-based heterogeneous particle swarm optimization and its application in uav communication coverage," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 3, pp. 312–323, 2020.

[12] J. Liu, S. Anavatti, and M. G. H. Abbass, "Comprehensive learning particle swarm optimisation with limited local search for uav path planning," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019, pp. 2287–2294.

[13] H. Wang, S. Lou, J. Jing, Y. Wang, W. Liu *et al.*, "The ebs-a* algorithm: An improved a* algorithm for path planning," *PLOS ONE*, vol. 17, no. 2, p. e0263841, 2022.

[14] R. Fareh, M. Baziyad, T. Rabie, I. Kamel, and M. Bettayeb, "Sobel potential field: Addressing responsive demands for uav path planning techniques," *Drones*, vol. 6, no. 7, p. 163, 2022.

[15] J. Chang, N. Dong, D. Li, W. H. Ip, and K. L. Yung, "Skeleton extraction and greedy-algorithm-based path planning and its application in uav trajectory tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 6, pp. 4953–4964, 2022.

[16] X. B. Yu, N. J. Jiang, X. M. Wang, and M. Y. Li, "A hybrid algorithm based on grey wolf optimizer and differential evolution for uav path planning," *Expert Systems with Applications*, vol. 215, p. 119327, 2023.

[17] Y. Shen, Y. L. Zhu, H. W. Kang, X. P. Sun, Q. Y. Chen, and D. Wang, "Uav path planning based on multi-stage constraint optimization," *Drones*, vol. 5, no. 4, p. 144, 2021.

[18] M. Y. Yu, J. Du, X. X. Xu, J. Xu, F. Jiang, S. W. Fu, J. Zhang, and A. K. Liang, "A multi-strategy enhanced dung beetle optimization for real-world engineering problems and uav path planning," *Alexandria Engineering Journal*, vol. 118, pp. 406–434, 2025.

[19] Y. Cui, W. Hu, and A. Rahmani, "Fractional-order artificial bee colony algorithm with application in robot path planning," *European Journal of Operational Research*, vol. 306, no. 1, pp. 47–64, 2023.

[20] M. Y. Yu, H. R. Yang, X. J. Wei, S. W. Fu, J. Xu, and J. Zhang, "Carbon footprint efficiency ratio: A unified indicator for green evaluation of evolutionary algorithms," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–14, 2026.

[21] X. B. Yu and W. G. Luo, "Reinforcement learning-based multi-strategy cuckoo search algorithm for 3d uav path planning," *Expert Systems with Applications*, vol. 223, p. 119910, 2023.

[22] Y. B. Cui, W. Hu, and A. Rahmani, "A reinforcement learning based artificial bee colony algorithm with application in robot path planning," *Expert Systems with Applications*, vol. 203, p. 117389, 2022.

[23] F. B. Wu, S. B. Li, J. X. Zhang, L. Y. Yu, X. Xiong, and L. B. Wu, "Q-learning-based exponential distribution optimizer with multi-strategy guidance for solving engineering design problems and robot path planning," *Results in Engineering*, vol. 27, p. 105705, 2025.

[24] H. R. Yang, M. Y. Yu, J. Q. Zhang, Y. F. Xiong, D. L. Wang, and J. Xu, "A reinforced quantum aquila optimizer for multi-threat 3d uavs path planning in complex environments," *Applied Mathematical Modelling*, vol. 155, p. 116736, 2026.

[25] L. Xu, X. B. Cao, W. B. Du, and Y. M. Li, "Cooperative path planning optimization for multiple uavs with communication constraints," *Knowledge-Based Systems*, vol. 260, p. 110164, 2023.

[26] Y. Liu, X. J. Zhang, Y. Zhang, and X. M. Guan, "Collision free 4d path planning for multiple uavs based on spatial refined voting mechanism and pso approach," *Chinese Journal of Aeronautics*, vol. 32, no. 6, pp. 1504–1519, 2019.

[27] M. Yu, H. Yang, J. Zhang, K. Ouyang, S. Fu, P. Tan, F. Jiang, and J. Xu, "Bounty hunter optimizer: A novel metaheuristic with an application to multi-uav mobile edge computing and path planning," *Knowledge-Based Systems*, vol. 341, p. 115836, 2026.

[28] P. Xing, H. Zhang, M. E. Ghoneim *et al.*, "Uav flight path design using multi-objective grasshopper with harmony search for cluster head selection in wireless sensor networks," *Wireless Networks*, vol. 29, pp. 955–967, 2023.

[29] S. H. Yin and Z. R. Xiang, "Energy-constrained collaborative path planning for heterogeneous amphibious unmanned surface vehicles in obstacle-cluttered environments," *Ocean Engineering*, vol. 330, p. 121241, 2025.

[30] Y. B. Niu, X. F. Yan, Y. Z. Wang, and Y. Z. Niu, "Three-dimensional collaborative path planning for multiple ucavs based on improved artificial ecosystem optimizer and reinforcement learning," *Knowledge-Based Systems*, vol. 276, p. 110782, 2023.

[31] Z. Xia, Y. Liu, J. Lu, J. Cao, and L. Rutkowski, "Penalty method for constrained distributed quaternion-variable optimization," *IEEE Transactions on Cybernetics*, vol. 51, no. 11, pp. 5631–5636, 2021.

[32] G. Gnecco, M. Gori, S. Melacci, and M. Sanguineti, "Learning with mixed hard/soft pointwise constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 9, pp. 2019–2032, 2015.

[33] W.-N. Chen *et al.*, "Particle swarm optimization with an aging leader and challengers," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 2, pp. 241–258, 2013.

[34] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the Fourteenth International Conference on Machine Learning (ICML)*, 1997, pp. 412–420.

[35] X. Li, "Niching without niching parameters: Particle swarm optimization using a ring topology," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, pp. 150–169, 2010.

[36] Q. Jiang and J. Li, "A novel method for finding global best guide for multiobjective particle swarm optimization," in *2009 Third International Symposium on Intelligent Information Technology Application*, 2009, pp. 146–150.

[37] Q. Yang *et al.*, "Random contrastive interaction for particle swarm optimization in high-dimensional environment," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 4, pp. 933–949, 2024.

[38] R. Zhong, J. Yu, X. Du, E. Zhang, and A. G. Hussien, "Improved competitive swarm optimizer with linear population reduction for large-scale optimization," in *2025 IEEE Congress on Evolutionary Computation (CEC)*, 2025, pp. 1–8.

[39] S. Liu, Z.-J. Wang, Y.-G. Wang, S. Kwong, and J. Zhang, "Bi-directional learning particle swarm optimization for large-scale optimization," *Applied Soft Computing*, vol. 149, p. 110990, 2023.

[40] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, 1999.

[41] J. Shahrabi, M. A. Adibi, and M. Mahootchi, "A reinforcement learning approach to parameter estimation in dynamic job shop scheduling," *Computers & Industrial Engineering*, vol. 110, pp. 75–82, 2017.

[42] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[43] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, pp. 1995–2003.

[44] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2016.

[45] Q. Yang, W.-N. Chen, T. Gu, H. Jin, W. Mao, and J. Zhang, "An adaptive stochastic dominant learning swarm optimizer for high-dimensional optimization," *IEEE Transactions on Cybernetics*, vol. 52, no. 3, pp. 1960–1976, 2022.

[46] R. Lan, Y. Zhu, H. Lu, Z. Liu, and X. Luo, "A two-phase learning-based swarm optimizer for large-scale optimization," *IEEE Transactions on Cybernetics*, vol. 51, no. 12, pp. 6284–6293, 2021.

[47] Y. Zhang, "Elite archives-driven particle swarm optimization for large scale numerical optimization and its engineering applications," *Swarm and Evolutionary Computation*, vol. 76, p. 101212, 2023.

[48] Z. Liu and T. Nishi, "Strategy dynamics particle swarm optimizer," *Information Sciences*, vol. 582, pp. 665–703, 2022.

[49] X. Xia *et al.*, "A particle swarm optimization with adaptive learning weights tuned by a multiple-input multiple-output fuzzy logic controller," *IEEE Transactions on Fuzzy Systems*, vol. 31, no. 7, pp. 2464–2478, 2023.

[50] G.-C. Ma, Q. Yang, J.-Y. Li, H. Zhao, X.-D. Gao, Z.-Y. Lu, and J. Zhang, "Tuple leading differential evolution for black-box optimization," *Expert Systems with Applications*, vol. 287, p. 128158, 2025.
[51] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou, "Chaotic local search-based differential evolution algorithms for optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3954–3967, 2021.
[52] Y. Yu, S. Gao, Y. Wang, and Y. Todo, "Global optimum-based search differential evolution," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 2, pp. 379–394, 2019.
[53] R. Zhong, A. G. Hussien, S. Zhang, Y. Xu, and J. Yu, "Space mission trajectory optimization via competitive differential evolution with independent success history adaptation," *Applied Soft Computing*, vol. 171, p. 112777, 2025.
[54] R. Zhong, A. G. Hussien, E. H. Houssein, and J. Yu, "Enhanced crested ibis algorithm: Performance validation in benchmark functions, engineering problems, and application in brain tumor detection," *Expert Systems with Applications*, vol. 289, p. 128231, 2025.
[55] X. Liu, T. Wang, Z. Zeng, Y. Tian, and J. Tong, "Three stage based reinforcement learning for combining multiple metaheuristic algorithms," *Swarm and Evolutionary Computation*, vol. 95, p. 101935, 2025.
[56] X. Wang, V. Snášel, S. Mirjalili, J.-S. Pan, L. Kong, and H. A. Shehadeh, "Artificial protozoa optimizer (apo): A novel bio-inspired metaheuristic algorithm for engineering optimization," *Knowledge-Based Systems*, vol. 295, p. 111737, 2024.
[57] J. Sun, X. Liu, T. Bäck, and Z. Xu, "Learning adaptive differential evolution algorithm from optimization experiences by policy gradient," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 4, pp. 666–680, 2021.