Manifold-Aware Triple Cooperative Multi-Population Differential Evolution with Reinforcement Learning for Irregular 3D UAV Path Planning

Yunhui Zhang [a], Guanglong Du [a*], Hao Tang [b], Ziwei Wang [c], Xueqian Wang [d], Cuifeng Du [e], Quanlong Guan [f], Xiaojian Qiu [g]

[a] School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510000, China

[b] School of Information and Communication Engineering, Hainan University, Haikou, 570000, China

[c] School of Engineering, Lancaster University, Lancaster LA1 4YW, United Kingdom

[d] Center of Intelligent Control and Telescience, Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, 518000, China

[e] CETC Potevio Science & Technology Co., Ltd., Guangzhou, 510000, China

[f] Research Institute for Guangdong Intelligent Education, Jinan University, Guangzhou, 510000, China

[g] Institute for Military-Civilian Integration of Jiangxi Province, Nanchang, 330000, China

*Corresponding author

E-mail addresses: csyunhuizhang@mail.scut.edu.cn (Yunhui Zhang), csgldu@scut.edu.cn (Guanglong Du)

Abstract This study proposes a manifold-aware triple cooperative multi-population differential evolution algorithm with reinforcement learning (MTCMDE) for UAV path planning in irregular three-dimensional environments with obstacle constraints and wind disturbances. The method aims to enhance both global exploration and local refinement, enabling more reliable identification of feasible low-cost paths in cluttered environments. MTCMDE adopts a three-subpopulation co-evolution framework consisting of Scout, Developer, and Balancer. These subgroups exchange information through elite migration, which helps balance exploration and exploitation during the search process. In addition, a manifold-aware perturbation operator is introduced to leverage local topological cues, reducing the risk of premature convergence and improving search efficiency in narrow passages. Furthermore, a reinforcement learning–driven multi-operator scheme based on a bandit model is employed to adaptively allocate mutation strategies, thereby improving convergence accuracy while maintaining population diversity. Experiments were conducted on six wind-affected irregular test cases, including a large-scale, ultra-high-density urban scenario, using a six-dimensional path cost function. The results show that MTCMDE achieves better path feasibility, lower cost, and faster convergence than the compared algorithms. Comprehensive statistical evaluations confirm the robustness and extreme

scalability of these improvements. Overall, MTCMDE provides an effective and scalable solution for UAV path planning in irregular 3D environments with wind disturbances.
Keywords UAV Path Planning; Differential Evolution; Multi-Population Cooperative Optimization; Manifold-Aware Search; Reinforcement Learning; Irregular 3D Environments

1. Introduction

The deployment of unmanned aerial vehicles (UAVs) has become increasingly integral to applications such as military reconnaissance [1], environmental monitoring [2], disaster response [3], logistics delivery [4], communication [5,6], and cloud computing [7]. A critical prerequisite for these missions is the generation of safe and cost-effective flight paths within complex 3D environments [8]. This task is particularly challenging because UAVs must navigate irregular terrains and dynamic threats while compensating for aerodynamic disturbances such as wind fields. These factors create a constrained, non-convex optimization landscape that poses extreme difficulties for conventional path planners.

To address this challenge, a range of heuristic and metaheuristic methods have been applied to UAV path planning, such as A* algorithm [9], rapidly-exploring random tree (RRT) [10], ant colony optimization (ACO) [11], particle swarm optimization (PSO) [12], genetic algorithms (GA) [13], and differential evolution (DE) [14]. In particular, DE is attractive owing to its concise structure, competitive global search ability, and amenability to parallel implementation. However, standard DE often suffers from premature convergence and limited adaptability to changing conditions, and its local refinement can be insufficient when dealing with cluttered 3D obstacles and complex terrain [15]. To alleviate these problems, recent studies have explored hybrid DE algorithms, adaptive parameter control, and learning-based strategies [16,17]. However, the scalability of these methods in real-world irregular environments is still insufficient, especially when UAVs need to pass through narrow gaps, steep terrain and turbulent airflow.

The primary bottleneck lies in the lack of coordination between exploration and exploitation mechanisms. Most existing frameworks apply static mutation operators across the entire population, disregarding local terrain information and specific role assignments. Furthermore, standard path cost models often underestimate the coupling effects of energy consumption and wind vectoring, thereby compromising the practical feasibility of the generated paths.

In summary, when performing complex low-altitude tasks, there exists a disconnect between the pure mathematical search behavior of standard heuristic algorithms and the

irregular 3D physical topological constraints faced by UAVs in practice. In real flight environments, including rugged terrain, dense asymmetric buildings, and dynamic wind disturbances, normally safe flight corridors become extremely narrow and tortuous [18]. However, most existing DE-based algorithms treat path planning as an unconstrained continuous optimization problem in Euclidean space. When faced with fragmented, non-convex obstacle spaces with highly irregular boundaries, algorithms performing stochastic linear mutations become trapped in local optima with severe Euclidean blind zones. This spatial blindness inevitably generates numerous infeasible trajectories that traverse obstacles or violate UAV kinematics. This not only leads to a catastrophic waste of computational power for fitness evaluation but also traps the population in high-penalty collision zones, causing irreversible local stagnation.

Inspired by these key bottlenecks, this research aims to reconstruct the evolutionary paradigm of UAV path planning in complex environments by injecting spatial geometric priors into the stochastic search. By introducing a manifold perception mechanism, the local geometric repulsion force of irregular obstacles is extracted. This enabled the UAV's search trajectory to smoothly glide along a collision-free topological submanifold, thus replacing blind trial and error in obstructed Euclidean space. High-quality actual flight paths must simultaneously consider global obstacle avoidance safety and local smoothness and cost reduction. The homogeneous populations equipped with static operators cannot simultaneously handle the global penetration of irregular and narrow manifolds and the fine development of local dynamic safety paths at the underlying structural level. This paper specifically constructs a triple cooperative heterogeneous multi-population architecture and entrusts reinforcement learning (RL) for real-time intelligent operator scheduling, thereby solving this highly conflicting irregular 3D optimization task.

To address these limitations, this paper proposes a novel algorithm named manifold-aware triple cooperative multi-population differential evolution with reinforcement learning (MTCMDE). The MTCMDE algorithm proposed in this paper is a differential evolution framework that integrates multi-subgroup co-evolution, memory-based adaptive parameter control, reinforcement learning-based policy scheduling, manifold-aware perturbation operators, and a multi-strategy library hybrid mechanism. The main innovations of this paper are as follows:

• A three-subgroup co-evolution framework is proposed, consisting of Scout, Developer, and Balancer roles to coordinate global exploration and local exploitation through periodic interaction and elite migration.

• A manifold-aware perturbation operator is designed to exploit local topological features, enhancing the ability to escape local optima and traverse narrow gaps in high-dimensional, irregular 3D environments.

• A reinforcement learning-driven multi-operator adaptation mechanism is developed, where a Bandit-based strategy selector dynamically allocates mutation operators from a role-guided evolution library integrating both classic and customized DE variants.

• A six-dimensional path cost model and a hybrid grid-based irregular obstacle modeling method are developed and benchmarked against eight state-of-the-art algorithms in six wind-affected irregular scenarios, including a large-scale, ultra-high-density urban scenario, demonstrating the superior performance of MTCMDE.

The rest of this paper is organized as follows. Section 2 reviews related work on UAV path planning and evolutionary optimization methods. Section 3 introduces the problem formulation and preliminary concepts. Section 4 presents the proposed MTCMDE algorithm. Section 5 reports and discusses the experimental results and comparisons, including validations in a large-scale ultra-high-density urban scenario. Finally, Section 6 concludes the paper.

2. Related work

2.1 Path planning in irregular 3D environments

Path planning for UAVs in irregular 3D environments has attracted extensive attention due to its high complexity and practical relevance. Traditional geometric and graph-based algorithms, such as A* and PRM, can generate feasible routes in structured maps but often fail to maintain efficiency and stability in unstructured or dynamic spaces [19,20]. These methods are sensitive to irregularities and prone to local stagnation when obstacle boundaries are non-convex or uncertain.

To address these issues, swarm intelligence algorithms have been introduced for optimization [21]. DE, PSO, and related metaheuristics exhibit strong global search capability and adaptability in multi-constraint optimization problems [22,23]. However, achieving both convergence precision and real-time performance in high-dimensional 3D planning remains a challenge.

Recent studies have further emphasized energy-intensive hybrid UAV path optimization frameworks. Cabreria et al. [24] proposed a grid-based energy minimization model for irregular regions. Choi et al. [25] developed a terrain-adaptive trajectory optimization method based on Gaussian modeling. These studies highlight the trend from static geometry to adaptive, hybrid, and intelligent driving strategies [26]. Salehi et al.[27] proposed the LAHC

algorithm for UAV path planning in complex urban environments with irregularly shaped threat areas. Irregularities may also stem from dynamic disturbances, such as wind fields or moving threats.

2.2 Differential evolution and variants

DE is a simple yet powerful continuous optimization evolutionary algorithm known for its robustness and fast convergence. However, its performance is highly dependent on control parameters such as scaling factor and crossover rate. To enhance adaptability, JADE [28] introduces adaptive control and external archiving, while SHADE and LSHADE [29,30] combine success history adaptation and linear population reduction to achieve a better balance between exploration and exploitation. LSHADE-SPA [31] improved convergence stability through semi-parameter adaptation, and SaUSDE [32] integrated multiple mutation strategies with self-adaptive selection to balance local and global searches.

In addition to DE variants, hybrid strategies such as the self-learning genetic algorithm (SLGA) [33] and comprehensive learning PSO with local search (CLPSO-LS) [34] have been proposed to enhance dynamic learning and multimodal search efficiency. Although these algorithms show superior performance on benchmark functions, their single-population structure limits adaptability of UAVs in complex 3D path planning tasks involving irregular obstacles and multi-constraint cost functions. Multi-population cooperation and adaptive hybridization remain key directions for advancing UAV path optimization.

2.3 Multi-population cooperative evolution strategies

To overcome the stagnation and diversity loss inherent in single-population evolutionary models, researchers have increasingly adopted multi-population cooperative evolution frameworks. These methods divide the population into subgroups with distinct roles or strategies to achieve heterogeneous co-optimization through interaction and information exchange. Representative approaches include the multi-island and coevolutionary frameworks [35], the cooperative multi-population DE variants [36,37], and hybrid tri-population or adaptive collaboration schemes [38,39]. This design enhances exploration capabilities by enabling different subgroups to pursue complementary search objectives while maintaining cohesion through elite migration and shared archives.

Recent studies have shown that dynamic role assignment and adaptive collaboration can further enhance global search and constraint handling capabilities [40–42]. However, most existing frameworks still rely on static communication or fixed migration intervals, limiting their responsiveness to dynamic environmental changes. In complex multi-objective or constrained optimization tasks, integrating adaptive role coordination and feedback-driven

policy assignment has become an emerging trend in balancing exploration and exploitation.

2.4 Reinforcement learning in evolutionary optimization

Recent studies have incorporated RL into evolutionary optimization to achieve adaptive control of operators, parameters, and resources. By framing the search process as a sequential decision problem, RL provides feedback-driven adaptation that improves algorithmic responsiveness and learning efficiency [43]. Among these, the multi-armed bandit (MAB) framework is notably effective at adaptive operator selection for DE and swarm-based algorithms owing to its simplicity and strong policy discrimination [44,45].

Subsequent studies have combined deep or multi-agent RL with evolutionary computation to achieve superior contextual awareness and decision stability. Martinez et al. [46] and Song et al. [47] applied RL to multitask and multi-objective settings. Choi et al. [48] and Arani et al. [49] demonstrated RL-driven UAV path planning and energy-efficient trajectory design. These works highlight the capability of RL to improve exploration–exploitation balance in dynamic environments.

Most RL and evolutionary algorithm (EA) hybrids remain limited to global population-level adaptation, lacking fine-grained subpopulation coordination or temporal awareness. Future research trends point toward hierarchical, context-aware RL-evolutionary frameworks for more adaptive and self-organizing optimization.

2.5 Manifold-guided search and topological awareness

Manifold-guided optimization has recently attracted attention for its ability to exploit intrinsic geometric structures in high-dimensional or constrained spaces. By embedding the search process into Riemannian or hyperbolic manifolds, these methods can capture curvature and continuity information, thus guiding populations more effectively through narrow or rugged regions [50,51]. Topology-aware frameworks further integrate geometric projection, gradient guidance, and cross-sectional pattern heuristics to enhance trajectory replanning and motion feasibility [52,53]. Moreover, manifold-based control and vector-field analysis provide robust convergence guarantees for path-following tasks in dynamic environments [54]. Despite these advances, existing studies mainly focus on local geometry or deterministic control, lacking adaptive perturbation control and population-level coordination.

Existing heuristic planners and their cutting-edge adaptive variants have made significant progress in the field of 3D UAV path planning. Nevertheless, they continue to exhibit shortcomings in their underlying mechanisms when faced with extremely complex, non-convex, and highly constrained real-world physical environments. To clearly define the innovative boundaries and technical barriers of this study, we provide an in-depth, structured

comparison of current mainstream classical baseline algorithms and the latest cutting-edge frameworks with the proposed MTCMDE. **Table 1** details the limitations of existing research and the targeted breakthroughs of MTCMDE from five core dimensions: population architecture, spatial search strategy, operator scheduling, environmental coupling, and trap escape capability.

Table 1 Multi-dimensional comparison of the proposed MTCMDE framework against existing algorithms.

No.	Algorithmic Dimension	Classical Baseline Algorithms	State-of-the-Art Algorithms	The Proposed MTCMDE Framework
1	Population architecture	Single, homogeneous population without role division.	Multi-population or external archive mechanisms, but mostly homogeneous in task execution.	Heterogeneous triple-subpopulation (Scout, Developer, Balancer) with explicit task division and high-frequency elite migration.
2	Spatial search strategy	Blind Euclidean stochastic perturbations.	Adaptive Euclidean step sizes based on historical success, lacking spatial topological awareness.	Manifold-aware topological perturbations. Injects geometric boundary priors to force trajectories to safely slide along narrow manifolds.
3	Operator scheduling	Static, manually tuned parameters fixed for all environments.	Dynamic self-adaptation of parameters, typically relying on delayed historical memory.	RL-driven (Bandit-based) real-time scheduling. Selects optimal mutation strategies instantly from a customized role-guided library.
4	Environmental coupling	Treats planning as generic unconstrained optimization; largely ignores aerodynamics.	Introduces dynamic penalty functions, but lacks fine-grained geometric environment mapping.	Deeply coupled with a hybrid grid-based obstacle model and a 6D path cost model (adding wind disturbance).
5	Trap escaping capability	Weak. Extremely prone to irreversible stagnation in non-convex collision zones.	Moderate. Escapes via random restarts or archived global bests, suffering from sluggish recovery.	Extremely robust. Achieves instantaneous topological broadcasts via manifold-repulsion and immediate subpopulation migration.

By combining the multidimensional benchmarking matrix in **Table 1**, it can be clearly seen that existing research is often limited by homogeneous population paradigms and blind Euclidean spatial perturbations. The limitations of this underlying architecture not only exacerbate the internal friction between algorithm exploration and development but also lead to catastrophic computational waste and irreversible stagnation of the population when crossing narrow and non-convex collision zones. In contrast, the core advantage of MTCMDE lies in its reconstruction of the optimization logic of traditional algorithms: it effectively breaks the homogenization bottleneck through the refined division of labor among

heterogeneous subgroups. By injecting geometric boundary priors (manifold perception), we achieved a paradigm transition from blind trial and error to topological guidance. The deep coupling of operator scheduling with 6D physical cost in conjunction with RL endows the algorithm with extremely strong local optimal escape and global convergence capabilities in extremely complex terrains. This structured comparison not only clarifies the inherent pain points of previous work but also provides theoretical motivation for the design of multi-dimensional collaborative frameworks proposed in this paper.

3. Preliminary

3.1 Problem formulation of 3D path planning

The core of the three-dimensional path planning problem lies in generating a feasible path $\Theta = \{P_1, P_2, ..., P_N\}$ for a flying object from a starting point P to a destination Q, where each control point $P_i \in \mathbb{R}^3$. The entire path should avoid the threat area Γ while achieving good performance across multiple performance metrics. This problem is typically modeled as a multi-objective optimization problem as follows:

$$\min_{\mathrm{P}} f(\Theta) = \sum_{i=1}^{M} \omega_i \cdot f_i(\Theta) \quad \text{s.t. } \mathrm{P} \cap \Gamma = \varnothing \tag{1}$$

where $f(\Theta)$ represents the performance cost of the path on the *i-th* evaluation indicator. $\omega_i \in [0,1]$ is the normalized weight of each indicator, satisfying $\sum_{i=1}^{M} \omega_i = 1$. Γ represents the set of threat bodies. $\mathrm{P} \cap \Gamma = \varnothing$ indicates that the path must completely avoid obstacles. M is the number of evaluation indicators.

As shown in **Fig. 1**, the UAV studied in this paper needs to avoid a large number of irregular polyhedron obstacles in 3D space and consider the dynamic impact of wind field interference.

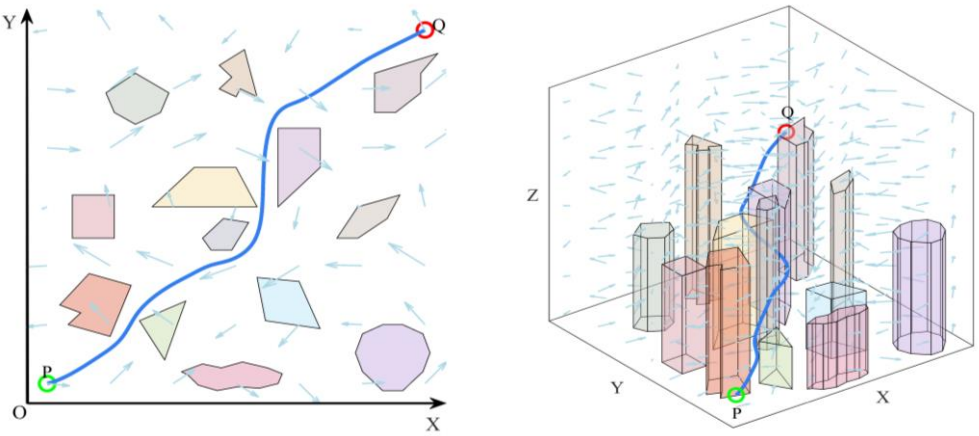


Fig. 1 UAV flight schematic in 2D and 3D.

To improve search efficiency in irregular 3D environments, a new local coordinate system is established by aligning the straight line from the start point P to the end point Q

with the X'-axis direction, as illustrated in **Fig. 2**. It represents a 2D projection perspective. The entire trajectory is then uniformly divided into $D+1$ segments, denoted by L_k where $K=0,\ 1,\ 2,...,\ D+1$. The coordinates of each path point are $x_k=(x_k,\ y_k,\ z_k)$, representing a continuous sequence of track points from the starting point x_0 to the end point x_{D+1}.

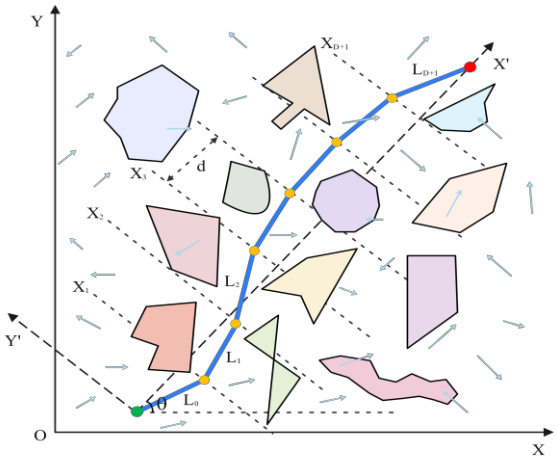


Fig. 2 Path planning under irregular obstacles.

Fig. 3 provides a 3D perspective, showing the path from P to Q as it passes through multiple planes. Using a slice modeling approach, a hybrid grid modeling method is established, where each plane can be considered a level in the path optimization dimension. This reflects the evolution of the path in the spatial dimension, requiring it to avoid obstacles at different heights while adapting to wind field disturbances to achieve irregular obstacle avoidance.

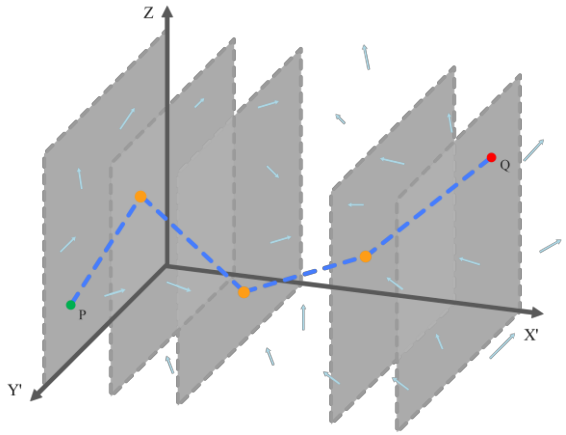


Fig. 3 Layered slice modeling.

3.2 Differential evolution algorithm

DE is an evolutionary optimization algorithm based on real number coding. It is widely used in nonlinear optimization problems due to its simple structure and strong global search capability.

In the path planning task, individuals are represented by real numbers, and each

individual is a sequence of path control points:

$$x_i^g = \left\{ x_{i,1}^g, x_{i,2}^g, ..., x_{i,D}^g \right\} \tag{2}$$

where $D = 3N$ represents N three-dimensional control points. g is the number of iterations. The algorithm mainly includes the following steps:

(1) Mutation

For each individual x_i^g in the population, randomly select three different individual indices $r1$, $r2$, $r3$ and generate a mutation vector:

$$v_i^g = x_{r1}^g + F \cdot \left(x_{r2}^g - x_{r3}^g \right) \tag{3}$$

where $F \in (0,2)$ is a scaling factor that adjusts the magnitude of the difference vector.

(2) Crossover

Use binary crossover strategy to generate test vector u_i^g:

$$u_{i,j}^g = \begin{cases} v_{i,j}^g, & if\ rand_j \le CR \text{ or } j = j_{rand} \\ x_{i,j}^g, & otherwise \end{cases} \tag{4}$$

where $CR \in [0,1]$ is the crossover probability.

(3) Selection

Compare the test individual with the original individual according to the fitness function and decide whether to replace it:

$$x_i^{g+1} = \begin{cases} u_i^g, & \text{if } f(u_i^g) < f(x_i^g) \\ x_i^g, & otherwise \end{cases} \tag{5}$$

3.3 Feasibility determination of paths

In 3D irregular environments, to ensure the feasibility and safety of UAV generated paths, the MTCMDE algorithm adopts a soft constraint mechanism based on minimum distance to handle irregular threats. This mechanism calculates the shortest distance between a waypoint and the surface of a threatening object and converts it into a continuous penalty term embedded in the integrated fitness function. This guides individuals away from obstacle areas during the evolutionary process.

(1) Modeling of irregular threat bodies

Each threat target is referred to as the j-th threat body Γ_j, which is modeled as a 3D cylinder formed by extending an arbitrary 2D polygonal base D_j in the vertical direction, which is defined as follows:

$$D_j = \left\{ (x,y,z) \middle| (x,y) \in D_j, z \in \left[z_j^{\min}, z_j^{\max} \right] \right\} \tag{6}$$

where $D_j = \left\{ (x,y) \in \mathbb{R}^2 \right\}$ represents the irregular boundary of the threat body on the

horizontal plane. $z \in \left[z_j^{\min}, z_j^{\max} \right]$ represents its vertical height range.

(2) Path discretization and sampling points

As mentioned earlier, a path consists of several control points $\Theta = \{\mathrm{p}_1, \mathrm{p}_2, ..., \mathrm{p}_N\}$, $\mathrm{p}_i = (x_i, y_i, z_i)$. Any two consecutive points form a path segment $L_i = [\mathrm{p}_i, \mathrm{p}_{i+1}]$. To approximate the spatial relationship between the path and the threat object, the algorithm selects representative sampling points (midpoint of the line segment) in each line segment:

$$m_i = \frac{p_i + p_{i+1}}{2}, \quad i = 1, 2, ..., N-1 \tag{7}$$

The spatial position of the sampling point m_i is used to calculate its shortest distance to the irregular threat body.

(3) Distance calculation model

For any sampling point p_i, first calculate the minimum 2D distance from point (x_i, y_i) to polygon D_j in the horizontal direction:

$$d_j(\mathrm{p}) = \min_{(\sigma,\upsilon)\in\delta D_j} \sqrt{(x_i - \sigma)^2 + (y_i - \upsilon)^2} \tag{8}$$

where δD_j is the polygonal boundary. It scans the polygon edge segments one by one, calculates the Euclidean distance from the point to each edge segment, and takes the minimum value. If (x_i, y_i) is already inside the polygon, it returns 0.

In the vertical direction, $d_j^{(Z)}(\mathrm{p})$ represents the distance that the point deviates from the height range of the threat body in the vertical direction, defined as:

$$d_j^{(Z)}(\mathrm{p}) = \max\left(0, \left| z - z_j^{mid} - \frac{h_j}{2} \right|\right) \tag{9}$$

where $z_j^{mid} = \left(z_j^{\min} + z_j^{\max}\right)/2$, $h_j = z_j^{\max} - z_j^{\min}$.

Combine the horizontal and vertical distances into a 3D minimum distance estimate to the threat body:

$$d_{j\,\mathrm{sum}}(\mathrm{p}) = \sqrt{\left[d_j(\mathrm{p})\right]^2 + \left[d_j^{(Z)}(\mathrm{p})\right]^2} \tag{10}$$

Take the minimum value of all threat bodies to get the closest threat distance of the point:

$$d_{\min}(\mathrm{p}) = \min_j d_{j\,\mathrm{sum}}(\mathrm{p}) \tag{11}$$

Fig. 4 shows a schematic diagram of the minimum distance from the path sampling point to the irregular threat body under two-dimensional projection.

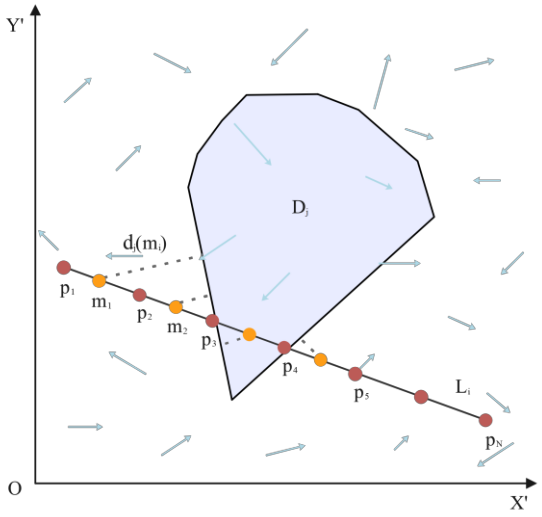


Fig. 4 Irregular threat distance evaluation in 2D projection.

In the MTCMDE algorithm, to enhance the feasibility and smoothness of the UAV's path, the resulting sequence of discrete path points is smoothly reconstructed using a cubic B-spline curve. As a parameterized curve with excellent continuity and local controllability, the cubic B-spline curve effectively eliminates sudden changes while preserving the overall path shape. It is suitable for multidimensional path optimization and reconstruction tasks in irregular 3D environments. A schematic diagram of the cubic B-spline curve is shown in **Fig. 5**.

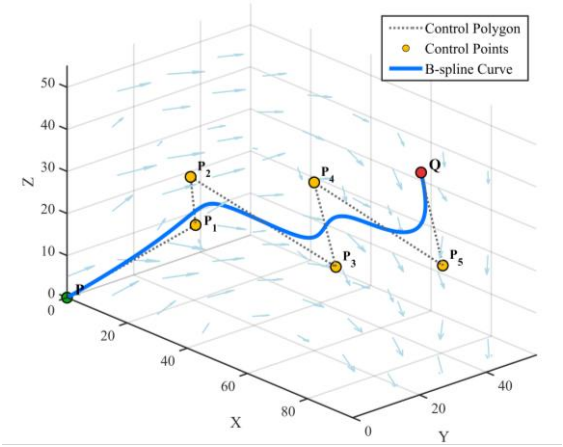


Fig. 5 Cubic B-spline curve.

3.4 Six-dimensional fitness function

(1) Path length cost

Path length cost f_{length} represents the cumulative length of the entire path segment and is used to measure the total flight distance. The path consists of N control points and the total length is defined as:

$$f_{\text{length}} = \sum_{i=1}^{N-1} \left\| \mathbf{p}_{i+1} - \mathbf{p}_i \right\|_2 \tag{12}$$

where $\| \cdot \|_2$ represents the Euclidean distance. This cost represents the cumulative length of the entire path segment and is used to measure the total flight distance.

(2) High deviation cost

High deviation cost f_{altitude} is used to penalize flying away from the ideal altitude.

$$f_{\text{altitude}} = \sum_{i=1}^{N-1}\left[\exp(0.02\cdot\left|z_{avg}^{(i)} - z_{ideal}\right|) - 1\right] \tag{13}$$

where $z_{avg}^{(i)} = (z_i + z_{i+1})/2$ represents the average altitude of the current path segment. z_{ideal} represents half the map height, representing the ideal flight altitude. This term is used to penalize flights that are far from the ideal altitude.

(3) Energy consumption cost

Energy consumption cost f_{energy} represents the energy consumption of the flight segment.

$$f_{\text{energy}} = \sum_{i=1}^{N-1}\left[\frac{1}{2}mv_i^2 + mg\cdot\max(0,\ z_{i+1} - z_i)\right] \tag{14}$$

where, $vi = \left\|p_{i+1} - p_i\right\|/dt$. $m = 2kg$ is the quality of the UAV. The first term is kinetic energy, and the second term is potential energy during the climb process.

(4) Corner smoothness cost

Corner smoothness cost f_{angle} encourages path turning to smoothness.

$$f_{\text{angle}} = \sum_{i=2}^{N-1} 5\cdot k_i^2,\ \ k_i = \frac{2\sin(\theta_i/2)}{\left\|\mathbf{v}_{i-1}\right\| + \left\|\mathbf{v}_i\right\|} \tag{15}$$

where θ_i is the angle between two adjacent path segments. k_i is the discrete curvature, which reflects the degree of path curvature.

(5) Safety cost

Safety cost f_{safety} strongly penalizes paths that are close to or cross obstacles. When the sampling point is too close to the threat body (less than the safety threshold d_{safe}), the path will be penalized exponentially:

$$\phi(d) = \sum_{i=1}^{N-1}\begin{cases} 10^4\left[1 - \exp(-\dfrac{d_{\text{safe}} - d}{10})\right], & d < d_{\text{safe}} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

$$f_{\text{safety1}} = \sum_{i=1}^{N-1}\phi(d_{\min}(m_i)) \tag{17}$$

If there is a control point on the path that actually falls into the threat body, that is, $p_i \in \Gamma_j$. Then a collision penalty term will be added:

$$f_{\text{safety2}} = 10^5\cdot N_{\text{collide}} \tag{18}$$

where N_{collide} represents the number of obstacles crossed in the path. Then the total safety

cost is:

$$f_{\text{safety}} = f_{\text{safety1}} + f_{\text{safety2}} \tag{19}$$

(6) Wind field interference cost

This study utilizes the wind field interference cost f_{wind} to improve the ability to resist wind interference.

$$f_{\text{wind}} = \sum_{i=1}^{N-1} \left[\lambda_1 \cdot \max(0, -\mathbf{v}_{\text{seg}} \cdot \mathbf{v}_{\text{wind}}) \cdot (\|p_{i+1} - p_i\|) + \lambda_2 \cdot \left\|\mathbf{v}_{\text{seg}} \times \mathbf{v}_{\text{wind}}\right\| \right] \tag{20}$$

where, λ_1=0.8, λ_2=0.1 denote wind direction coefficients. $\mathbf{v}_{\text{seg}}$ is the direction vector of the path segment. $\mathbf{v}_{\text{wind}}$ represents the wind speed vector at the midpoint. The preceding term represents the windward cost, while the latter term represents the crosswind cost.

To achieve comprehensive optimization of UAV paths in 3D irregular environments, the MTCMDE algorithm designs a six-dimensional composite fitness function to measure the comprehensive performance of the path in various aspects. The objective function is defined as:

$$f(\Theta) = \omega_1 \cdot f_{\text{length}} + \omega_2 \cdot f_{\text{altitude}} + \omega_3 \cdot f_{\text{energy}} + \omega_4 \cdot f_{\text{angle}} + \omega_5 \cdot f_{\text{safety}} + \omega_6 \cdot f_{\text{wind}} \tag{21}$$

where $\omega_i \in [3,\ 0.5,\ 0.25,\ 1,\ 8,\ 0.5]$ is the weight coefficient. After normalization, the sum of ω_i is 1.

4. Proposed MTCMDE algorithm

4.1 Algorithm framework

To effectively address the problem of UAV path planning in irregular 3D environments, this paper constructs a MTCMDE algorithm. It integrates multi-subgroup collaborative evolution, manifold-aware perturbation, and reinforcement learning-based policy scheduling.

MTCMDE divides global exploration and local development tasks into three subgroups: Scout, Developer, and Balancer. Through periodic interactions and elite migration, it achieves heterogeneous collaboration, enhancing the diversity and stability of the search. In terms of strategy selection, it introduces a context-aware multi-armed slot machine model that divides the evolutionary process into multiple stages. Based on fitness feedback, it dynamically adjusts the invocation probability of mutation strategies to achieve stage-aware adaptive strategy scheduling. For high-dimensional obstacle environments, a manifold-aware perturbation operator is designed to integrate local topological structure and subgroup role information to enhance individual cross-obstacle search capabilities. The algorithm integrates multiple conventional and improved mutation strategies to construct a multi-strategy hybrid library, which can be flexibly invoked based on subgroup role, further improving overall

convergence and adaptability.

To facilitate the understanding of the working mechanism of MTCMDE, its flowchart is presented in **Fig. 6** and the corresponding pseudocode is provided in **Algorithm 1**.

Algorithm 1: The pseudo code of **MTCMDE** algorithm

Input:

N (population size), D (dimension), *MaxGen* (maximum generations), *gridMap* (3D environment), *threats* (irregular obstacles), start and goal positions, *nPoints* (control points), *params* (algorithm parameters)

Output:

BestPath (optimal path), *costHistory* (fitness convergence history), *stats* (statistical performance indicators)

Generate three role-based subpopulations: $\{\text{ScoutPop, DevPop, BalPop}\}$ via intelligent initialization strategy, and initialize RL bandit states // *Initialize roles and Q-tables*
Evaluate all individuals using the path cost function and identify the global best solution X_{best}
for generation $t = 1$ to *MaxGen* **do**
    for each subpopulation $S \in \{\text{Scout, Developer, Balancer}\}$ and each individual $X_i \in S$ **do**
        Select DE operator op_id via RL-based bandit mechanism // *RL-driven operator selection*
        Generate mutant solution via the selected operator; apply manifold-aware perturbation when the manifold-based operator is selected // *Inject manifold-aware geometric priors*
        Apply crossover to produce a trial solution X_{trial}
        Evaluate $f(X_{\text{trial}})$ using the path cost function // *6D complex path cost evaluation*
        if $f(X_{\text{trial}}) < f(X_i)$ **then**
            Replace $X_i \leftarrow X_{\text{trial}}$ and assign a positive reward to op_id // *Reward successful mutation*
            if $f(X_i) < f(X_{\text{best}})$ **then**
                Update $X_{\text{best}} \leftarrow X_i$ // *Update global optimum*
            end if
        else
            Assign a zero reward to op_id
        end if
        Update the RL bandit statistics for op_id
    end for
    Perform elite preservation and cross-subpopulation injection // *Periodic inter-population migration*
    if $t > 0.2 \times MaxGen$ **then**
        Gradually reduce subpopulation sizes according to the population control strategy
    end if
    if population diversity drops below a predefined threshold **then**
        $\{\text{ScoutPop, DevPop, BalPop}\} \leftarrow$ intelligent restart mechanism // *Trigger local optimum escape*
    end if
    Record the fitness value of X_{best} in $costHistory(t)$
end for
Apply path smoothing to obtain the final path *bestPath*
Compute statistical indicators

Return: $bestPath \leftarrow X_{\text{best}}$, *costHistory*, *stats*

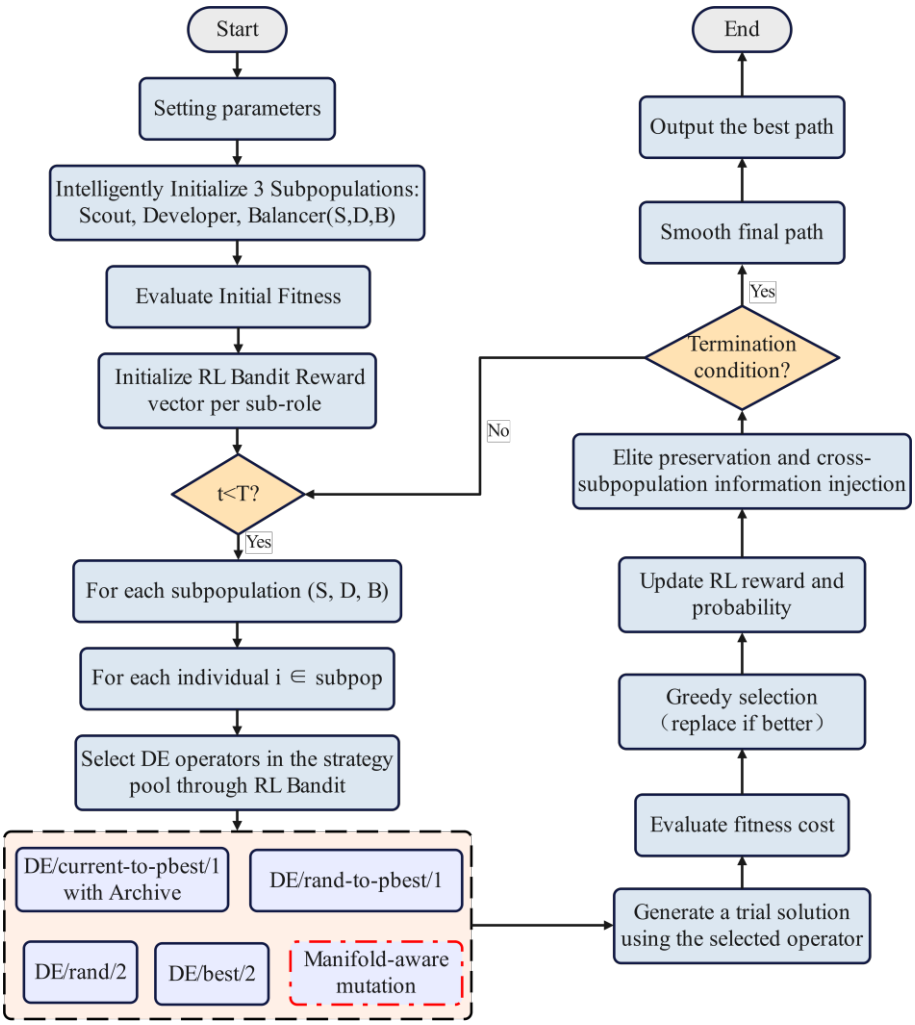


Fig. 6 Flowchart of the MTCMDE algorithm.

4.2 Multiple subgroups division of labor and collaboration

The MTCMDE algorithm adopts a parallel collaborative framework with three subgroups that play different search roles. The Scout subgroup is responsible for global exploration in the early evolutionary stage. The Developer subgroup focuses on local exploitation and solution refinement in the middle and later stages. The Balancer subpopulation acts as an information bridge between exploration and exploitation. It facilitates strategy transfer and helps maintain population diversity.

To enhance cooperation among these subgroups, a cyclical elite injection mechanism is employed. During each generation of evolution, elite individuals are transferred in a circular structure. In particular, elites from the Developer, Balancer, and Scout subgroups are injected into the Scout, Developer, and Balancer subgroups, respectively. This design enables continuous information exchange and provides effective guidance for convergence.

During evolution, the total population size adaptively adjusts and gradually decreases to 50% of its initial size to improve computational efficiency. The population is divided into

three role-based subgroups: Scout, Developer, and Balancer, in a ratio of approximately 0.4:0.4:0.2. Due to their different roles, these subgroups exhibit different search behaviors and operator preferences. The manifold-aware mutation operator is adaptively selected through a reinforcement learning-based mechanism, enabling the algorithm to adjust its search strategy based on evolutionary feedback.

During elite injection, a small number of elite individuals replace selected individuals in the target subgroup. The fitness values of the updated individuals are then re-evaluated. This process supports stable knowledge transfer across subgroups and helps preserve population diversity. **Fig. 7** illustrates the ring-based collaborative evolution framework of the Scout, Developer, and Balancer subgroups in MTCMDE.

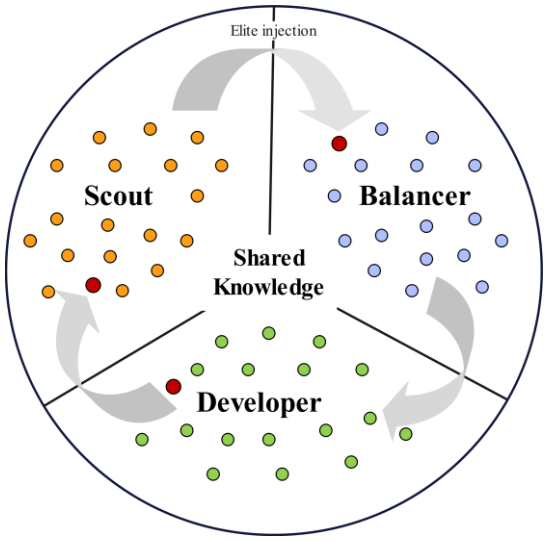


Fig. 7 Ring-based collaboration of three subgroups in MTCMDE.

4.3 Contextual bandit-type reinforcement learning policy selection

A multi-armed bandit model based on contextual state is introduced, dividing the evolutionary process into early, mid, and late stages. The optimal mutation strategy is dynamically selected based on the current stage. This process uses an ε-greedy strategy to balance exploration and exploitation, and uses normalized fitness gains as immediate rewards to iteratively update the strategy value function, implementing a stage-aware strategy scheduling mechanism.

Using evolution stages as contextual states $s \in \{1,2,3\}$. Early stage $p \le 0.3$, middle stage $0.3 < p \le 0.7$, late stage $p > 0.7$ .The action $a \in \{1,2,3,4,5\}$ consists of five operators. An ε-greedy strategy is adopted for operator selection:

$$a = \begin{cases} \text{Uniform}\{1,...,5\}, & \varepsilon \\ \arg\max_a Q(s,a), & 1-\varepsilon \end{cases} \tag{22}$$

After applying operator a to individual x_i, a trial solution ρ_i is obtained and evaluated. The immediate reward is defined as:

$$r = \frac{J(x_i) - J(\rho_i)}{J(x_i) + \epsilon} \tag{23}$$

where $J(\cdot)$ denotes the composite fitness value and ϵ is a small constant to avoid numerical instability.

The action-value function is updated as:

$$Q(s,a) \leftarrow Q(s,a) + \gamma\left(r - Q(s,a)\right) \tag{24}$$

where $\gamma \in (0,1]$ a is the learning rate. $Q(s,a)$ is the action value function, which measures the expected reward obtained by performing action a in state s. **Fig. 8** illustrates the workflow of the contextual bandit-type reinforcement learning policy selection in MTCMDE.

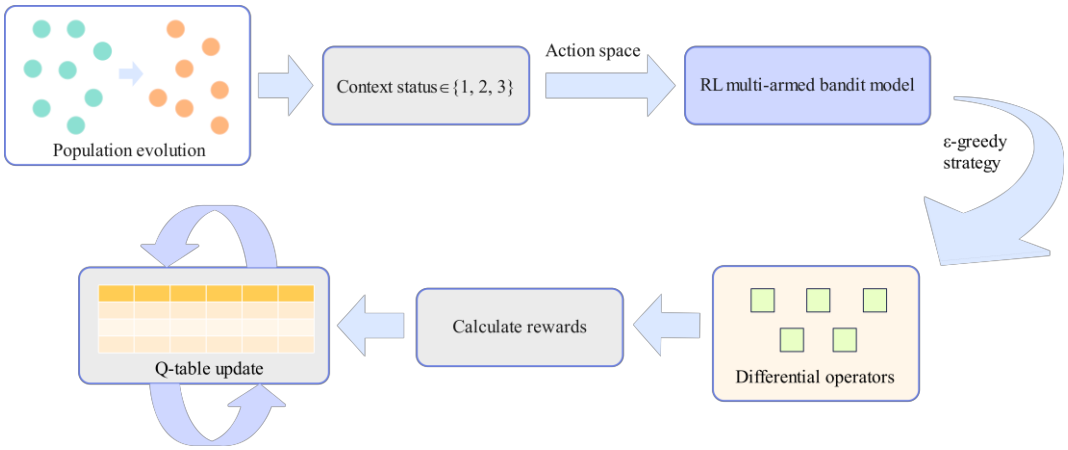


Fig. 8 Contextual bandit-based policy selection in MTCMDE

4.4 Manifold-aware mutation

To address the possible structure of feasible manifolds in complex, high-dimensional search spaces, we designed an adaptive perturbation mechanism that combines evolutionary stages with subgroup roles. This strategy builds on the basic DE/rand/1 operation and introduces a perturbation intensity parameter that varies with the number of evolutionary generations, allowing for differentiated designs for different subgroups. The Scout subgroup applies large perturbations to enhance global search capabilities. The Developer subgroup uses small perturbations to improve local search accuracy. The Balancer subgroup introduces structural perturbations in the form of sinusoidal functions to enhance the ability to traverse narrow channels and multi-obstacle bottlenecks.

The core theoretical motivation of this manifold-aware operator is as follows: In a non-convex and height-constrained 3D environment, the feasible solution space for safe

obstacle avoidance is not a regular Euclidean space. Instead, it manifests as an extremely narrow and fragmented topological manifold. Traditional differential mutation, with its blind random walks within Euclidean space, yields numerous invalid collision solutions. Therefore, this operator extracts the geometric repulsion force of irregular obstacle boundaries as a spatial constraint prior. This is equivalent to injecting a topological pseudo-gradient into the gradient-free heuristic algorithm. This mechanism forces the randomly evolving search trajectory to slide on the feasible manifold surface, mathematically transforming blind trial and error into a topologically guided directed exploration. Its mathematical principal expression can be derived as follows:

Based on $v_0 = x_{r1} + F \cdot (x_{r2} - x_{r3})$, adjust perturbations according to the evolutionary progress $\eta = t / \max Gen$ and subgroup roles:

$$\xi_{\text{explore}} = \alpha \cdot (1 - \eta),\ \xi_{\text{exploit}} = \beta \cdot \eta \tag{25}$$

where α, β denote the maximum exploration and exploitation intensity coefficients, with values of 1 and 0.5. Assign different update methods to different subgroups:

$$v_i = \begin{cases} v_i + \xi_{\text{explore}} \cdot \mathrm{U}[-\frac{1}{2}, \frac{1}{2}], & \text{Scout} \\ v_i + \xi_{\text{exploit}} \cdot \mathrm{U}[-\frac{1}{2}, \frac{1}{2}], & \text{Developer} \\ v_i + \left[(\xi_{\text{explore}} + \xi_{\text{exploit}}) / 2 \right] \cdot \sin(\tau v_0), & \text{Balancer} \end{cases} \tag{26}$$

where τ is the scaling factor, with a value of 1.5. Then, a binomial cross is used by row (intermediate node) while keeping the endpoints unchanged and grabbing out-of-bounds.

Current manifold-aware operator is highly dependent on calculating the geometric distance from control points to static obstacle boundaries. If the environment is highly dynamic or changes rapidly, the topology of feasible manifolds can change drastically in real time. This is the relative limitation of the MTCMDE algorithm, and we objectively position it as a robust offline global planner operating in static or quasi-static complex environments.

4.5 Multi-strategy library hybrid mechanism

The algorithm integrates a diverse library of five DE strategies, including:

(1) DE/current-to-pbest/1 with archive

$$v_i = x_i + F \cdot (x_{pbest} - x_i) + F \cdot (x_{r1} - x_{r2}) \tag{27}$$

where x_i is the current individual. x_{pbest} is randomly selected from the top p% of elite individuals. F is the scaling factor. x_{r1} is randomly sampled from the current population. x_{r2} is selected from a historical archive that stores recently replaced individuals.

(2) DE/current-to-pbest/1

$$v_i = x_i + F \cdot (x_{pbest} - x_i) + F \cdot (x_{r1} - x_{r2}) \tag{28}$$

where all individuals x_{pbest}, x_{r1} and x_{r2} are sampled from the current population without using the archive.

(3) DE/rand/2

$$v_i = x_{r1} + F \cdot (x_{r2} - x_{r3}) + F \cdot (x_{r4} - x_{r5}) \tag{29}$$

where $x_{r1}, x_{r2}, x_{r3}, x_{r4}, x_{r5}$ are five mutually distinct individuals randomly selected from the current population.

(4) DE/best/2

$$v_i = x_{best} + F \cdot (x_{r1} - x_{r2}) + F \cdot (x_{r3} - x_{r4}) \tag{30}$$

where x_{best} is the optimal individual in the current population.

(5) Manifold-aware mutation

In addition to classical DE operators, a manifold-aware mutation is incorporated to better exploit the geometric structure of feasible paths. This operator adaptively perturbs intermediate control points while keeping the start and goal points fixed. The perturbation intensity is adjusted according to evolutionary stage and subgroup role, as described in Section 4.4.

For all strategies, a dimension-wise binomial crossover is applied to intermediate control points of the path representation. Any out-of-bound solutions are projected back into the feasible search space using a clamping function, ensuring path feasibility and numerical stability. By combining these complementary mutation strategies under a reinforcement learning-driven selection mechanism, MTCMDE achieves a balanced trade-off between exploration and exploitation across different evolutionary stages.

4.6 Computational complexity analysis

The computational complexity of MTCMDE fundamentally depends on the path fitness evaluation process. Let *G* be the maximum number of iterations, *P* be the total population size, *N* be the number of control points for each path, *M* be the number of obstacles in the environment, and *V* be the average geometric complexity of each obstacle. To clearly quantify the additional overhead introduced by the advanced components proposed in this paper, the time complexity is broken down as follows:

1. RL-based operator selection: Bandit reward calculation and Q-table update only involve basic scalar arithmetic operations for each individual, with an algorithmic overhead of only $O(1)$.

2. Three-subgroup co-evolution: Periodic elite migration requires sorting the fitness arrays of each subpopulation, resulting in a sorting complexity of $O(P \log P)$.

3. Manifold-aware mutation: Calculating the local topological repulsion vector requires measuring the distance from the perturbed point to the nearest obstacle boundary. The worst-case upper bound of this step is completely consistent with standard collision detection, resulting in a complexity of $O(N \cdot M \cdot V)$.

The complexity of standard difference mutation and crossover operations is $O(N)$. Therefore, the six-dimensional fitness evaluation constitutes the absolute computational bottleneck, with an operational scale of $O(N \cdot M \cdot V)$. Because this higher-order term completely dominates the RL policy update ($O(1)$) and elite sorting ($O(P \log P)$) in an asymptotic sense, the overall time complexity of MTCMDE is strictly limited to $O(G \cdot P \cdot N \cdot M \cdot V)$. The space complexity primarily stems from the memory required to store the population trajectory matrix and historical archives. Because the population maintains P individuals, and each individual contains N control points in a 3D coordinate system, the overall space limit of the algorithm is strictly $O(P \cdot N)$.

The introduction of manifold sensing, multi-population cooperation, and reinforcement learning mechanisms did not lead to an increase in algorithm complexity. On the contrary, by mathematically filtering out a large number of redundant evaluations within infeasible collision regions, these modules effectively optimize the actual runtime.

5. Experimental results and analysis

5.1 Experimental setup

To verify the performance and robustness of the proposed MTCMDE algorithm for UAV path planning in irregular 3D environments, this study selected eight representative differential evolution and swarm intelligence algorithms as comparison objects. These include: classical differential evolution algorithm DE , adaptive differential evolution algorithm JADE, adaptive differential evolution algorithm SHADE based on success history, L-SHADE introducing linear population reduction mechanism, L-SHADE-SPA using semi parametric adaptive strategy, adaptive uniform search differential evolution algorithm SaUSDE, genetic algorithm SLGA integrating social learning mechanism, and particle swarm optimization algorithm CLPSO-LS combining comprehensive learning and local search. These algorithms cover various representative methods such as classic evolutionary algorithms, champion algorithms, and the latest improved evolutionary algorithms, and can comprehensively evaluate the comprehensive optimization performance of MTCMDE. All algorithms are applied to UAV path planning tasks in 3D irregular obstacle environments under the same constraints. The maximum iteration number is set to 800, the population size is 200, the number of control points is 20, the safe distance is 30, and the ideal flight altitude is set to 200.

Each algorithm runs independently 20 times to obtain stable statistical results.

The experimental environment was MATLAB R2020a, and the computing platform configuration was: Intel(R) Core (TM) Ultra 9 285H processor (2.9 GHz), 32 GB of memory, and Windows 11 64-bit operating system. All algorithms used consistent boundary conditions and termination criteria, and the parameter configuration is detailed in **Table 2**.

Table 2 Parameter settings of the compared algorithms.

Algorithms	Parameter Settings
DE	Mutation factor F : 0.4; Crossover rate Cr : 0.9
JADE	p-best ratio p : 0.2; Learning rate c : 0.1; Archive size: 100; Mean of F F_{mu} : 0.4; Mean of CR $\mathrm{Cr}_{\mathrm{mu}}$: 0.4
SHADE	p-best ratio p : 0.2; Archive size: 100; Memory size: 5; Mean of F F_{mu} : 0.4; Mean of Cr Cr_{mu} : 0.4
LSHADE	Minimum population: 4; Memory size: 5; Archive size: 400; Memory update c : 0.1; p-best ratio p : 0.2; Cauchy scale of F F_{Scale} : 0.1; Normal sigma of Cr Cr_{Sigma} : 0.1
LSHADE-SPA	Minimum population: 4; Memory size: 5; Archive size: 400; p-best ratio p : 0.2; Memory update c : 0.1; Fixed F F_{fixed} : 0.5; Fixed CR Cr_{fixed} : 0.9; Cauchy scale of F F_{Scale} : 0.1; Normal sigma of CR Cr_{Sigma} : 0.1
SaUSDE	Mutation factor F : 0.6; Min crossover $Cr_{\min}$: 0.4; Max crossover $Cr_{\max}$: 0.95
SLGA	Elite ratio p : 0.1; Learning rate α : 0.75; Discount factor λ : 0.2; Epsilon-greedy ε : 0.85; States: 20; Actions 10; Weight $\omega_1 = \omega_2$: 0.35; Weight ω_3 : 0.3; Crossover prob P_c : 0.4-0.9; Mutation prob P_m : 0.01-0.21
CLPSO-LS	Inertia weight ω : 0.6; Acceleration coefficient c : 1.5; Max velocity ratio $v_{MaxRatio}$: 0.2; Template refresh gap: 7; Tournament size: 5; Local-search top-K: 8; LS Gaussian sigma: 0.05; Stagnation window : 25
MTCMDE	Mutation factor F : 0.6; Crossover rate Cr : 0.85; Elite ratio p : 0.1; Restart threshold: 0.05; Improvement threshold: 1e-5; Variance threshold: 1e-4; Late-phase ratio: 0.95; Late improvement threshold: 0.001

5.2 Case description

To evaluate MTCMDE under challenging 3D path planning conditions, five simulation cases (Cases 1-5) were designed. All cases share the same environment size of 1000×1000×1000 to keep the comparison consistent and ensure that the results are reproducible. Each case combines irregular obstacles with wind-field disturbances, and the overall difficulty is increased progressively by making the scene more cluttered and the feasible region more constrained. Thus, these tests were designed to examine robustness from several viewpoints, including global search behavior, path feasibility, obstacle avoidance, and sensitivity to dynamic perturbations.

For a fair comparison, the start and finish positions are placed in relatively open areas while still being constrained by surrounding threats. This setting guarantees that feasible paths exist while avoiding trivial solutions. The typical configuration of each case is summarized in **Table 3**. Case 1 adopts a horizontal-view scenario with a constant flight altitude, serving as a baseline feasibility test. Cases 2, 4, and 5 use an upward-view setting to mimic takeoff or climbing trajectories, whereas Case 3 adopts a downward-view setting to highlight stability

during descent and landing. Across Cases 1-5, the number of obstacles increases from 11 to 40, moving from low-density to high-density environments and providing a more stringent assessment under wind disturbance.

The obstacle set is intentionally diverse and includes a range of complex geometries (e.g., cylindrical and pyramidal shapes, irregular polyhedra, bent and composite structures). The layouts vary from sparse to dense and from relatively regular patterns to more scattered distributions, which introduces strong nonlinearity and uncertainty into the search space. These obstacle configurations were generated to reflect practical mapping and challenging operating conditions, and were visualized using both 2D top-down views and 3D plots. The parameters are detailed in **Appendix A**.

Table 3 Typical settings of test cases.

Case	Number of obstacles	Start point	Goal point	Flight attitude
Case 1	11	[50,50,200]	[950,950,200]	Horizontal view
Case 2	17	[50,50,10]	[950,950,400]	Upward view
Case 3	22	[30,30,500]	[950,950,50]	Downward view
Case 4	30	[150,50,150]	[950,950,500]	Upward view
Case 5	40	[50,50,100]	[950,950,600]	Upward view

5.3 Convergence analysis

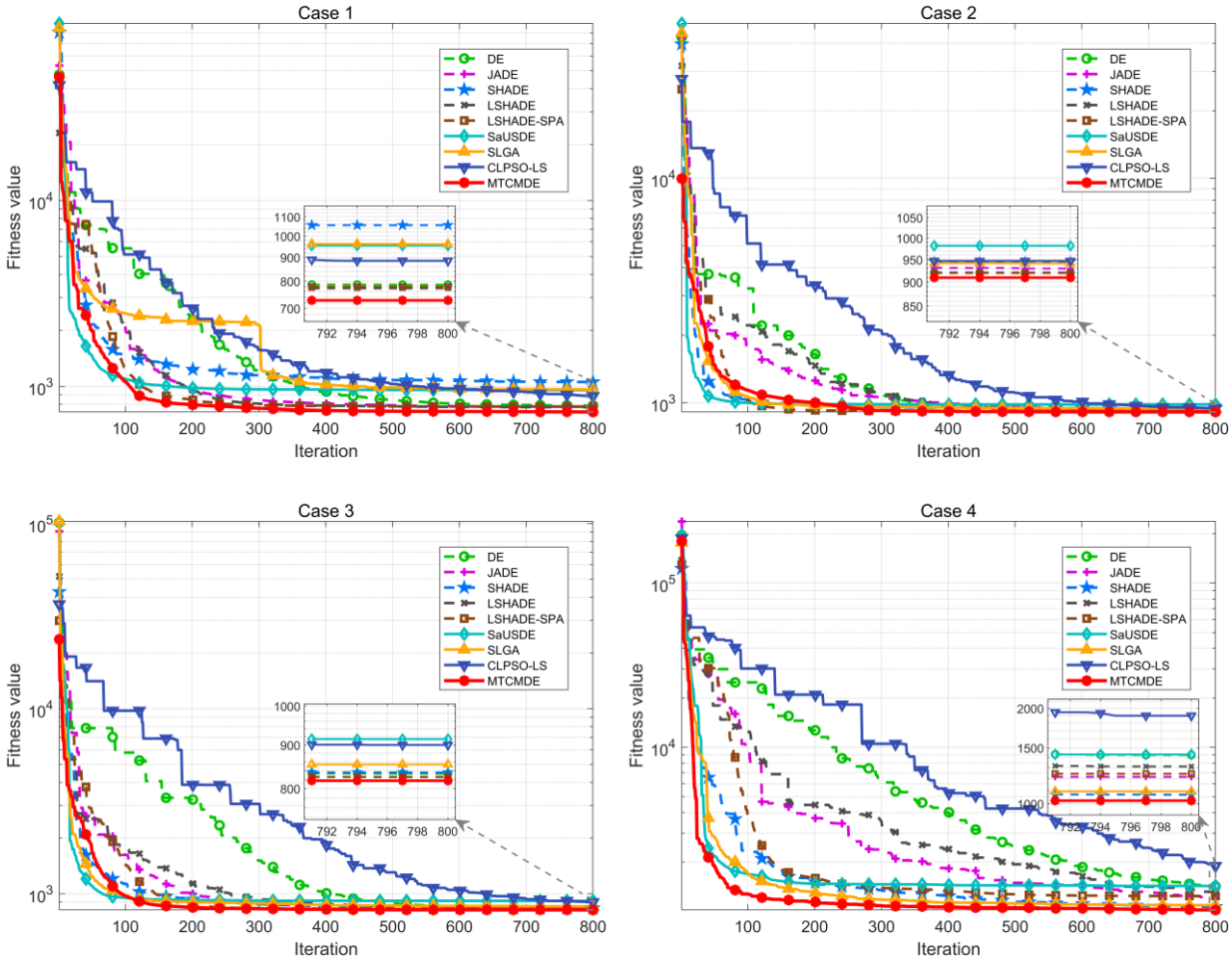
Fig. 9 compares the convergence behaviors of MTCMDE with eight baseline algorithms. In the first 100 iterations, the fitness values of most algorithms rapidly decreased, reflecting their global exploration behavior in the early stages of the search. The MTCMDE algorithm showed faster convergence speed in all five test scenarios, quickly approaching high-quality areas. This rapid decline is mainly attributed to the synergistic effect of its three subgroup structures and adaptive operator selection mechanism, which effectively improves early population diversity and global search capability.

In the mid to late stages, some comparative algorithms, such as JADE, LSHADE, and SLGA, experienced varying degrees of premature stagnation and slower changes in fitness. However, MTCMDE maintained a sustained downward trend in fitness in Cases 3 and 5, demonstrating its strong local exploration ability and stable search process. This indicates that the algorithm not only maintains a leading position in low complexity environments but also demonstrates strong adaptability in highly complex scenarios characterized by high-dimensional, non-convex, and wind disturbances.

A closer inspection of the final iterations (zoomed inset) reveals that MTCMDE secures the lowest asymptotic fitness with the most compact convergence curve. This minimal

variance underscores its superior stability and precision, enabling the identification of better global optima within a fixed computational budget. In comparison, SHADE and CLPSO-LS display noticeable oscillations, suggesting sensitivity to stochastic perturbations in the late search phase, whereas SaUSDE and SLGA yield suboptimal accuracy due to mid-stage stagnation.

Overall, MTCMDE displays an ideal convergence profile for trajectory optimization: rapid initial descent, sustained mid-term refinement, and stable, high-precision termination. Such behavior confirms its exceptional balance between global exploration and local exploitation, validating its efficacy for complex, multi-constrained path planning in irregular 3D terrains.



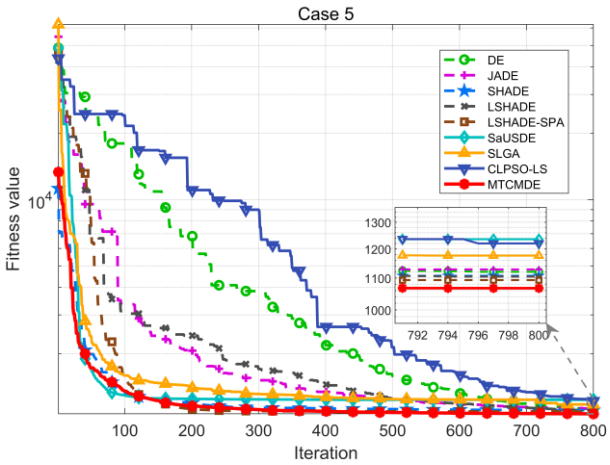


Fig. 9 Convergence curves of MTCMDE and comparison algorithms.

Table 4 presents the quantitative performance metrics across the five test cases. MTCMDE demonstrates distinct superiority regarding convergence precision, convergence rate, and stability. In terms of mean fitness, the proposed algorithm consistently achieves optimal or near-optimal rankings; this is particularly evident in Cases 2, 4, and 5, where it yields substantial improvements, recording values of 941.72, 1071.61, and 1128.96, respectively. Such results surpass the majority of comparative benchmarks, validating the robust global search mechanisms and adaptability of MTCMDE to problem constraints. Furthermore, MTCMDE achieves a steep descent in fitness within the initial 100 generations, marking a decisive improvement over traditional formulations like DE and JADE. The standard deviation of MTCMDE is the lowest or close to the lowest in all cases, indicating its highly stable convergence performance. Comprehensive analysis shows that MTCMDE can not only quickly enter the high-quality solution domain and obtain the optimal solution but also maintain minimal fluctuations and stable optimization results. Therefore, it is highly suitable for complex, multi constraint 3D path planning tasks.

Table 4 Statistical results of different algorithms for five cases.

Case	Statistic	DE	JADE	SHADE	LSHADE	LSHADE_SPA	SaUSDE	SLGA	CLPSO_LS	MTCMDE
	Best	786.31	776.49	1055.21	772.37	778.16	954.05	960.85	885.29	**727.91**
Case1	Avg.	816.11	804.87	1163.55	**799.42**	886.21	1171.00	998.32	923.18	804.07
	Sd.	19.60	20.69	86.90	**12.64**	103.82	64.00	153.53	24.21	43.01
	Best	920.59	929.81	944.60	921.02	919.86	981.88	940.97	946.84	**909.75**
Case 2	Avg.	1006.66	955.12	950.48	954.20	957.23	1032.04	991.16	1021.89	**941.72**
	Sd.	120.86	10.00	9.49	12.67	15.66	39.26	27.10	99.16	**7.59**
	Best	832.74	825.75	835.52	825.20	825.73	914.44	854.00	900.07	**817.16**
Case 3	Avg.	913.41	847.29	899.19	855.43	888.95	1014.08	933.69	963.71	**837.23**
	Sd.	63.37	23.77	39.06	33.83	30.31	72.32	57.12	53.46	**20.12**

	Best	1425.01	1214.99	1067.55	1310.72	1242.71	1430.01	1093.50	1894.81	**1022.42**
Case 4	Avg.	2003.03	1410.13	1105.19	1624.17	1989.66	1904.44	1526.42	2136.65	**1071.61**
	Sd.	417.35	111.24	87.23	158.04	598.15	449.45	567.60	256.05	**43.02**
	Best	1120.84	1128.83	1107.49	1105.00	1093.39	1235.65	1176.50	1220.14	**1067.11**
Case 5	Avg.	1862.23	1182.81	1159.12	1180.67	1155.65	1360.86	1235.85	1382.24	**1128.96**
	Sd.	1464.01	40.71	**22.76**	38.20	47.59	67.99	40.93	114.58	32.10

*The optimal results are shown in bold.

5.4 Path analysis in irregular cases

Figs. 10 and **11** illustrate the 3D path diagram and its top view, respectively. In all five cases, MTCMDE produces feasible paths that avoid the obstacle field, and its advantage becomes more evident when the scene is both cluttered and affected by wind. In Case 4, where obstacles are densely packed and geometrically diverse, several methods (e.g., DE and SHADE) yield less reliable behaviors, such as excessive detours or trajectories that cut through narrow corridors and may lead to obstacle penetration. MTCMDE tracks the obstacle boundaries more carefully, avoids collisions and abnormal deviations, and remains effective in compact feasible regions.

The paths generated by MTCMDE also show smaller curvature variations and fewer sharp turns than those produced by SaUSDE and SLGA. In Cases 4 and 5, it maintains smooth transitions while passing multiple adjacent non-convex obstacles, whereas CLPSO-LS and LSHADE-SPA often introduce abrupt heading changes that could complicate practical flight control. Under wind disturbances, MTCMDE preserves a stable heading trend and trajectory consistency, without the unstable oscillations observed in some baselines. In addition, its paths are generally shorter and more directionally efficient. In Cases 1 and 2, the planned route stays close to the lower boundary of the feasible space and reaches the goal with a lower overall cost. Furthermore, the top-down views obtained for MTCMDE show a coherent and compact navigation pattern through the multi-obstacle layout, whereas the algorithms used for comparison produce relatively more scattered or overly conservative routes.

MTCMDE demonstrates strong feasibility, smooth geometry, wind robustness, and flexible obstacle avoidance in irregular 3D environments. The resulting paths achieve lower costs and exhibit better engineering practicality and safety for complex constrained path planning tasks.

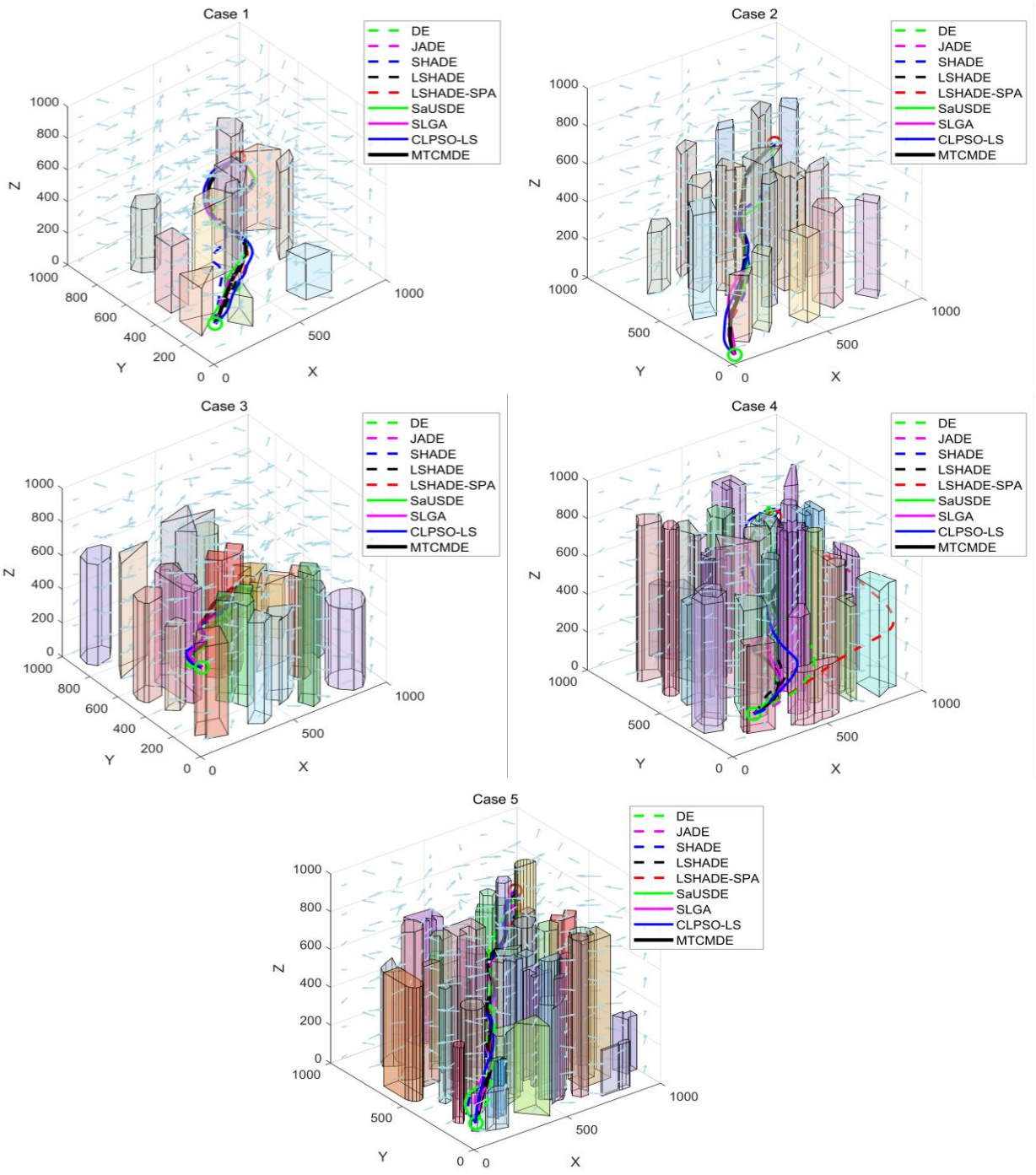


Fig. 10 3D path planning results in irregular environments under wind fields.

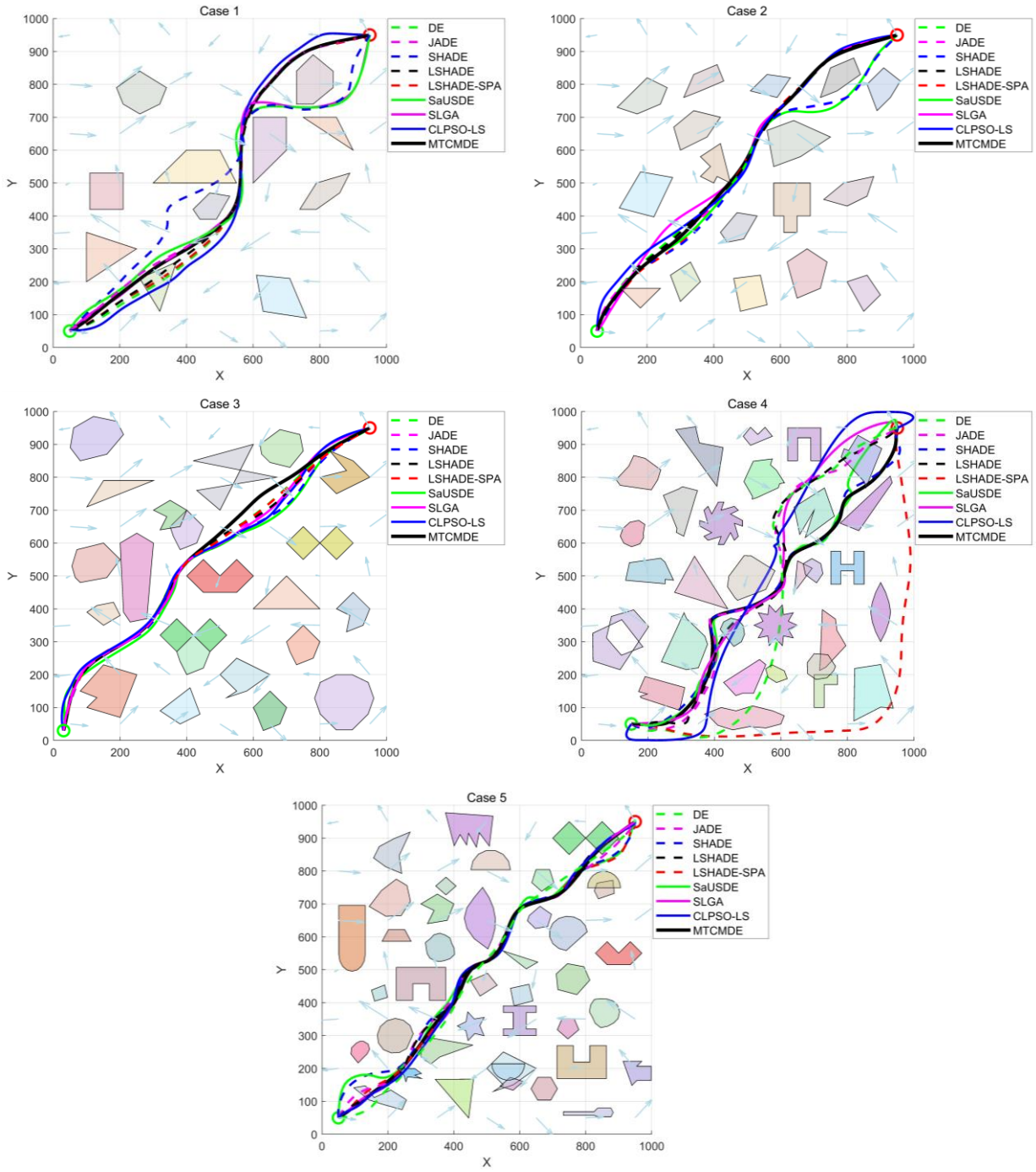


Fig. 11 Top-down view of the planned paths in irregular environments.

5.5 Runtime analysis

The running time line chart plotted in **Fig. 12** shows significant differences in the running times of various algorithms among the five irregular 3D path planning tasks. Overall, the traditional DE algorithm has the longest running time in all cases, especially Cases 3 and 4, with running times of 1897.02 s and 1690.66 s respectively, far exceeding those of other algorithms, indicating lower computational efficiency. In contrast, MTCMDE maintains a moderate running time in all test scenarios, averaging at approximately 628.61 s, far lower

than those of algorithms such as CLPSO-LS and SaUSDE. This is particularly evident in highly complex cases (Cases 3 and 4), where MTDMDE achieves running times of 673.35 s and 895.27 s, respectively, far lower than those of DE and SaUSDE. These findings demonstrate the outstanding time management capabilities of MTCMDE.

Although LSHADE and LSHADE-SPA achieve shorter runtimes in some scenarios, their computational advantages are limited by their disadvantages in path accuracy and convergence stability. While maintaining its leading comprehensive optimization quality, MTCMDE's runtime has not increased significantly, demonstrating a good balance between performance and efficiency. It achieved excellent runtime control, demonstrating the engineering practicality of its algorithm's multi-subpopulation parallel evolution and adaptive operator scheduling.

As noted in this computational complexity analysis, although the high-level components introduce minimal internal overhead, they fundamentally accelerate the overall search process by preventing subpopulations from wasting significant fitness evaluations in highly penalized collision zones. This theoretical mechanism closes the loop and explains the observed experimental results. To emphasize, MTCMDE achieves significantly shorter and more stable runtimes than traditional methods in highly constrained scenarios.

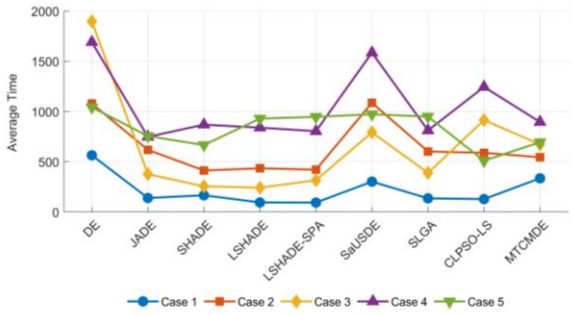


Fig. 12 Runtime comparison of different algorithms.

5.6 Statistical tests

5.6.1 Wilcoxon rank-sum test

To verify the scientific validity and reliability of the proposed MTCMDE algorithm, a Wilcoxon rank sum test was used for statistical significance analysis. This nonparametric test uses p-values to determine whether differences between algorithm runs are statistically significant [55,56].

The experimental results in **Table 5** show that, in most irregular environments, the p-values between MTCMDE and the comparison algorithms were far below the conventional significance level of 0.05. Especially, when compared with algorithms such as SHADE and

SaUSDE, the p-values were generally lower than powers 6 or even 8, indicating a highly statistically significant performance difference. This indicates that MTCMDE has significant advantages in path convergence quality, optimization stability, and robustness.

Although some individual cases did not reach low significance levels, the overall Wilcoxon test results showed a desirable significance trend. This proves that, MTCMDE exhibits relatively more reliable and superior optimization capabilities in most complex and irregular 3D environments. These findings validate its effectiveness and superiority in UAV path planning.

Table 5 Wilcoxon rank-sum test results between MTCMDE and comparison algorithms.

Case	MTCMDE VS							
	DE	JADE	SHADE	LSHADE	LSHADE-SPA	SaUSDE	SLGA	CLPSO-LS
Case1	3.50E-02	6.75E-02	6.79E-08	2.25E-02	1.95E-03	5.32E-08	1.60E-05	8.86E-05
Case 2	1.14E-02	1.04E-06	7.90E-08	2.04E-05	1.25E-05	6.79E-08	9.74E-06	6.79E-08
Case 3	9.75E-06	1.84E-02	1.26E-05	4.89E-02	3.07E-06	1.63E-03	4.54E-07	2.46E-04
Case 4	6.76E-08	6.71E-08	9.61E-03	9.17E-08	9.46E-08	1.22E-03	3.97E-06	4.60E-04
Case 5	7.11E-03	3.39E-03	3.04E-04	7.76E-02	1.63E-02	1.57E-06	3.96E-03	4.54E-06

5.6.2 Friedman test

To evaluate the comprehensive performance across all test cases, we employed the Friedman statistical test based on average fitness scores and relative rankings [57,58]. As detailed in **Table 6**, MTCMDE consistently achieved either the premier or secondary rank across the five scenarios. This performance underscores both its stability and competitive global optimization capability. While LSHADE exhibited a marginal edge over MTCMDE in Case 1, MTCMDE dominated the remaining four cases, evidencing superior robustness and broad applicability to irregular 3D path planning. In contrast, standard DE and other improved variants demonstrated marked volatility in performance, resulting in fluctuating rankings. Furthermore, algorithms such as SaUSDE and CLPSO-LS frequently occupied the lower tiers, signaling their limited adaptability to the specific constraints tested. Collectively, the Friedman statistics corroborate that MTCMDE maintains a distinct advantage in terms of both scoring metrics and ranking stability within complex 3D environments.

Table 6 Friedman test results of different algorithms.

Case	Indicators	DE	JADE	SHADE	LSHADE	LSHADE-SPA	SaUSDE	SLGA	CLPSO-LS	MTCMDE
Case1	Mean score	3.65	2.75	8.10	2.45	4.50	8.40	6.75	5.90	2.50
	Rank	4	3	8	1	5	9	7	6	2
Case 2	Mean score	4.35	4.55	3.15	4.70	5.00	8.20	6.75	6.45	1.85

Case 3	Rank	4	5	2	6	7	9	8	3	1
	Mean score	5.25	2.90	5.15	2.70	4.80	8.55	6.40	7.45	1.80
Case 4	Rank	6	3	5	2	4	9	7	8	1
	Mean score	7.05	4.05	1.80	5.25	6.75	6.80	4.40	7.50	1.40
Case 5	Rank	8	4	2	5	6	7	3	9	1
	Mean score	6.45	4.30	3.70	4.15	2.80	8.10	5.95	7.95	1.60
	Rank	7	5	3	4	2	9	6	8	1

5.6.3 Violin plot analysis

Fig. 13 visualizes the fitness distributions and solution densities across the five test cases via violin plots [59]. MTCMDE consistently secured the lowest or near-lowest mean fitness values. More importantly, its distribution profile was tightly clustered with negligible volatility and an absence of significant outliers, characteristics that underscore its exceptional convergence stability and robustness. This superiority was particularly pronounced in the highly complex environment of Case 4, where MTCMDE outperformed competing methods by a substantial margin, thereby validating its global optimization capability and adaptability to intricate 3D constraints. In sharp contrast, traditional DE paradigms and their variants displayed broad dispersion ranges and unstable solution densities across multiple scenarios, revealing their inherent limitations when navigating irregular topologies.

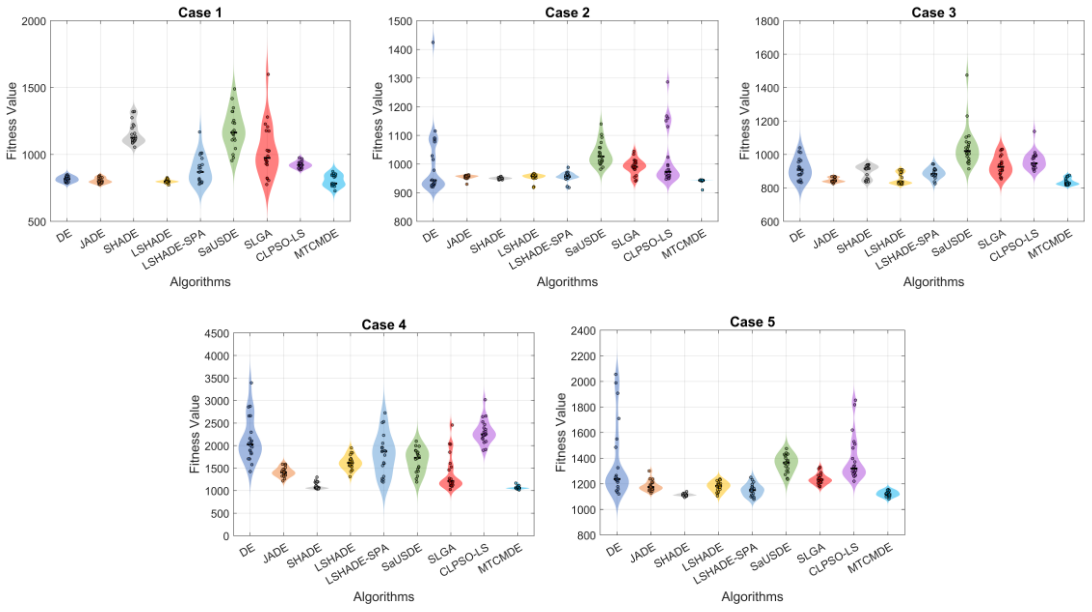


Fig. 13 Violin plots of fitness distributions for different algorithms.

5.6.4 Radar chart analysis

The radar chart distributions [60] plotted in **Fig. 14** show that the fitness values of MTCMDE in the five test cases always remained within the minimum range, significantly outperforming other algorithms. This demonstrates its high degree of convergence and

stability. In contrast, traditional DE and its variants fluctuated significantly in various dimensions, especially in Case 4, where the peak is prominent, reflecting their performance degradation in complex scenarios. Algorithms such as SaUSDE, SLGA, and CLPSO-LS showed a large radius distribution as a whole, indicating their obvious disadvantages in multiple scenarios. Overall, MTCMDE proved to be significantly advantageous within the convergence range of all dimensions, verifying its robustness and wide adaptability in irregular 3D path planning tasks.

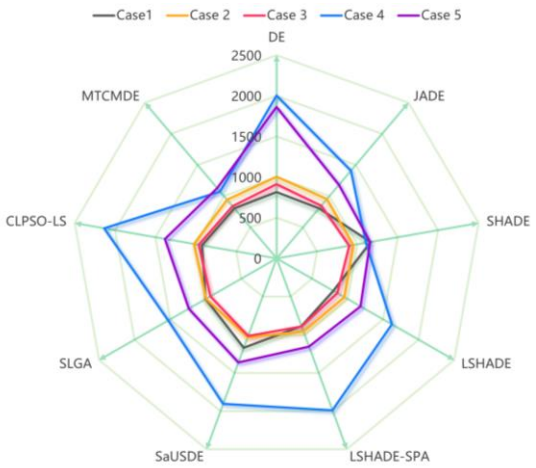


Fig. 14 Radar chart comparison of different algorithms' performance.

5.6.5 Effect Size Analysis

To further quantify the magnitude of performance differences beyond statistical significance, we calculated Cohen's effect size based on the raw fitness results from 20 independent runs. The complete results are summarized in **Table 7**.

According to statistical guidelines, an effect size of $d \approx 0.2$ is considered a small effect, $d \approx 0.5$ a moderate effect, and $d \geqslant 0.8$ a large effect. As detailed in **Table 7,** in all scenarios, MTCMDE exhibited a significant effect relative to a majority of compared algorithms. In highly constrained scenarios (such as Case 4), MTCMDE achieved a relatively large effect compared to baseline algorithms such as DE and JADE. These findings prove its strong practical significance in deterministic 3D path planning. This significance stems directly from the extremely low variance of MTCMDE and its ability to consistently avoid high-penalty collision zones, while traditional algorithms often stagnate in these local optima.

In the simpler Case 1, LSHADE slightly outperformed MTCMDE, producing a small negative effect (-0.15). This is entirely consistent with the Friedman ranking, objectively reflecting a clear algorithmic trade-off. MTCMDE prioritizes the development of complex nonconvex topologies over convergence speed in simple open spaces. Overall, the effect size confirms that MTCMDE's advantage is significant both statistically and practically.

Table 7 Cohen's d effect sizes for the fitness comparison between MTCMDE and other algorithms.

Case	MTCMDE VS							
	DE	JADE	SHADE	LSHADE	LSHADE-SPA	SaUSDE	SLGA	CLPSO-LS
Case1	0.36	0.02	5.24	-0.15	1.03	6.73	1.72	3.41
Case 2	0.76	1.51	1.02	1.20	1.26	3.20	2.48	1.14
Case 3	1.62	0.46	1.99	0.65	2.01	3.33	2.25	3.13
Case 4	3.14	4.01	0.49	4.77	2.16	2.61	1.13	5.80
Case 5	0.71	1.47	1.08	1.47	0.66	4.36	2.91	3.01

5.7 Sensitivity Analysis of Elite Migration Frequency

In the co-evolutionary framework, migration frequency determines the trade-off between subpopulation diversity and information sharing. To validate the configuration of elite migration intervals, we conducted a sensitivity analysis using a highly constrained case (Case 4). We evaluated algorithms with different migration intervals (intervals = 1, 5, 10, and 20) and compared them with an algorithm without migration. The statistical results based on 20 independent runs are summarized in **Table 8**.

The data show that the proposed high-frequency migration (interval = 1) performs best, achieving the lowest average path cost (1071.61) and standard deviation (43.02). The performance of algorithm significantly decreased with increasing interval. While classic unconstrained optimization typically employs delayed migration (interval = 10) to prevent premature convergence, the optimization environment of 3D UAV path planning has unique characteristics. In dense environments filled with irregular obstacles, feasible spatial pathways are extremely narrow and scarce. Once a specific subpopulation successfully discovers a collision-free manifold topology, this crucial spatial information must be shared immediately. Any migration delay forces other subgroups to waste significant fitness evaluations in infeasible collision regions, leading to severe convergence stagnation and high penalties. Therefore, setting the migration interval to 1 serves as an immediate topology broadcasting mechanism, which is theoretically sound and empirically optimal in complex non-convex environments. This is far more important than traditional diversity maintenance when solving complex 3D obstacle collision avoidance problems.

Table 8 Sensitivity analysis of elite migration frequency on MTCMDE performance in Case 4.

Migration Strategy	Migration Interval	Avg.	Sd.
MTCMDE (Proposed)	1	1071.61	43.02
MTCMDE-M5	5	1145.34	82.51
MTCMDE-M10	10	1318.72	175.85

MTCMDE-M20	20	1624.21	245.64
MTCMDE-NoMig	-	1910.55	382.30

5.8 Scalability Validation in a Large-Scale Real-World Application

To further demonstrate the scalability and universality of the application of the proposed method to UAVs in real-world scenarios, a large-scale, ultra-high-density urban scenario was constructed (Case 6). Unlike previous test cases, this environment spans a vast 2000 m × 2000 m × 500 m airspace and contains 80 highly irregular, non-convex obstacles. These obstacles are randomly distributed to realistically simulate the topological complexity of a modern central business district (CBD). The extreme obstacle density and intricate topological traps (such as deep concavities and lightning-shaped structures) severely compress the feasible flight manifold. This use case is specifically designed to rigorously evaluate whether the algorithm can maintain high robustness and successfully escape the Euclidean blind zone under large-scale spatial constraints.

Due to the exponential increase in computational cost in this scenario, we selected the highly representative algorithm DE and the high-performance algorithms LSHADE-SPA and CLPSO-LS for benchmark comparison. All settings in this experiment are consistent with previous experiments. The start and end points were set to (50, 50, 20) and (1950, 1950, 300).

Experimental results show a significant performance gap between the tested methods. The 3D trajectory and 2D top views (**Fig. 15**) show that the baseline DE algorithm cannot avoid dense obstacle clusters and will collide with building boundaries due to the inherent Euclidean blindness of the standard mutation operator. LSHADE-SPA and CLPSO-LS can bypass most obstacles, but when trapped in narrow manifolds between adjacent tall buildings, they exhibit significant trajectory detours and jagged flight segments. LSHADE-SPA performs relatively better. Nevertheless, MTCMDE outperforms all these algorithms by successfully identifying the optimal and smooth collision-free path. This advantage confirms that the manifold-aware perturbation operator can effectively utilize local geometric cues, enabling the UAV to glide along feasible manifold boundaries instead of getting stuck in dead ends.

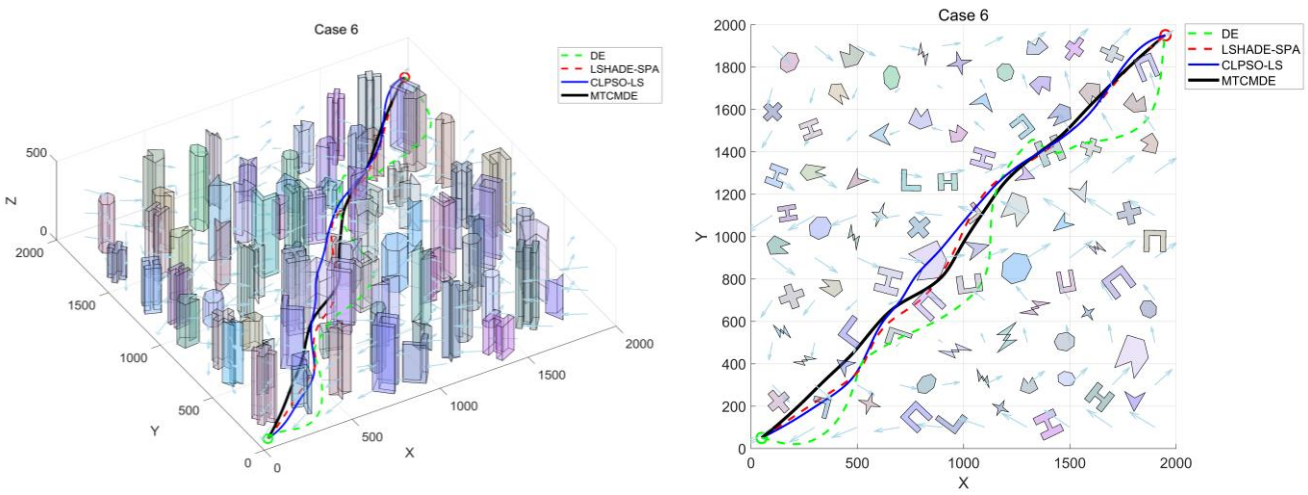


Fig. 15 Experimental results of 3D and 2D UAV irregular path planning in the large-scale, high-density urban environment (Case 6).

The convergence behavior shown in **Fig. 16** further highlights the scalability of the MTCMDE. In contrast, DE and CLPSO-LS exhibit insufficient convergence. LSHADE-SPA converges prematurely after approximately 500 iterations due to the rapid loss of population diversity in the large-scale constraint space. Whereas, MTCMDE maintains a continuous downward trend throughout the evolutionary process. The reinforcement learning-driven allocation scheme dynamically adjusts the policy pool, successfully avoiding search stagnation. Furthermore, as shown in **Table 9**, MTCMDE achieves the lowest average cost and smallest standard deviation in this large-scale experiment. These results demonstrate that the proposed framework not only possesses strong large-scale spatial scalability but also exhibits excellent generalizability for complex real-world path planning tasks.

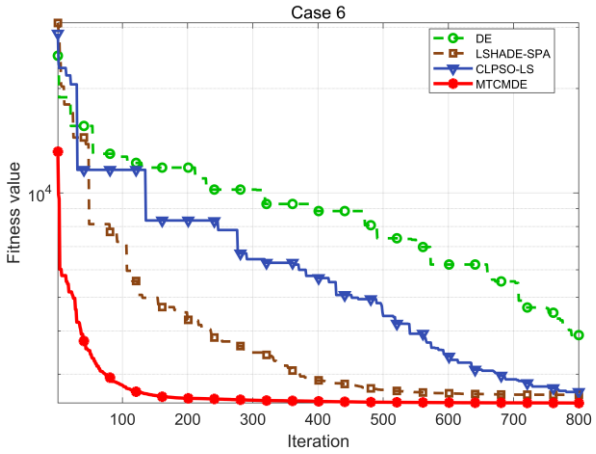


Fig. 16 Comparison of convergence curves of various evaluation algorithms in the large-scale urban scenario.

Table 9 Statistical performance comparison of various algorithms in the large-scale scenario (Case 6).

Algorithm	Best	Avg.	Sd.
DE	3888.94	6983.42	1418.11
LSHADE-SPA	2616.98	3058.52	395.19
CLPSO-LS	2666.42	3593.40	789.89
MTCMDE	2478.34	2561.45	52.29

6. Conclusion

This paper proposes a novel DE framework (MTCMDE) for 3D UAV path planning in the presence of wind field disturbances and irregular obstacles. It integrates the collaboration of three sub-populations (Scout/Developer/Balancer), multi-operator adaptation driven by Bandit reinforcement learning, and manifold-aware disturbance. Moreover, it supports a six-dimensional path cost model and irregular obstacle grid modeling. In systematic experiments comparing eight representative algorithms across six complex wind-affected cases, including an ultra-high-density urban scenario with 80 obstacles, MTCMDE can stably generate feasible, shortest, and smoother 3D trajectories, showing faster early convergence and continuous late improvement, with lower final fitness cost and variance. Path visualization, violin plots, and radar plots demonstrate balanced performance across multiple metrics, including obstacle avoidance safety, angle and height smoothness, and wind field robustness. Statistical tests, including the Wilcoxon rank-sum test, the Friedman test, and Cohen's d effect size evaluations, further support that the observed performance gains are significant and consistent. Coupled with a rigorous computational complexity analysis, while MTCMDE does not consistently minimize runtime metrics, it yields superior trajectory quality in high-complexity scenarios. This behavior reflects a pragmatic trade-off between computational expenditure and solution fidelity. The proposed framework is characterized by rapid convergence, robust search stability, and precise localization of optima. The distinct adaptability of MTCMDE to irregular topologies establishes it as a rigorous methodology for UAV path generation subject to intricate constraints.

Future work will extend MTCMDE to multi-objective optimization, utilizing reinforcement learning to balance Pareto trade-offs [61–64]. To address the current limitation regarding the dependence on real-time topological changes of feasible manifolds in highly dynamic environments, this framework will be expanded to multi-UAV cooperative planning with dynamic collision avoidance strategies [65,66]. Furthermore, the integration of Internet of Things technologies[67,68] will be explored to enhance real-time situational awareness and distributed decision-making capabilities for UAV clusters.

CRediT authorship contribution statement

Yunhui Zhang: Writing - Original Draft, Methodology, Software, Formal analysis, Visualization. **Guanglong Du:** Writing - Review & Editing, Conceptualization, Resources, Supervision, Project administration, Funding acquisition. **Hao Tang:** Writing - Review & Editing, Validation, Investigation. **Ziwei Wang:** Formal analysis, Data Curation, Software. **Xueqian Wang:** Writing - Review & Editing, Validation. **Cuifeng Du:** Investigation, Visualization. **Quanlong Guan:** Data Curation, Software. **Xiaojian Qiu:** Resources, Supervision, Writing - Review & Editing

Data Availability

Data will be made available on request.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Guangdong Provincial Natural Science Foundation for Outstanding Youth Team Project (Grant No. 2024B1515040010); the TCL Charity Foundation Young Scholars Project (Grant No. 2024001); the Guangdong Basic and Applied Basic Research Foundation (Grant Nos. 2023B1515120064 and 2023A1515011248); the Dreams Foundation of Jianghuai Advanced Technology Center (Grant No. 2023ZM01Z013); the Open Foundation of Jiangxi Province Key Laboratory of Military-Civilian Integration Beidou Navigation (Grant Nos. 2024JXRH0Y03 and 2024JXRH0Y05); the National Key R&D Program of China (Grant Nos. 2023YFC3603800, 2023YFC3603804, and 2023YFC3603803); the Shenzhen Medical Research Fund (Grant No. C2501005); the Guangdong Major Project of Basic and Applied Basic Research (Grant No. 2025B0303000013); and the Fundamental Research Funds for the Central Universities.

AppendixA. Details of irregular obstacles (Cases 1-5)

Case 1

No.	Projection coordinates and height of obstacles	No.	Projection coordinates and height of obstacles
1	(100, 350), (100, 200), (250, 300) 300	2	(320, 110), (370, 260), (260, 210) 180
3	(750, 700), (850, 700), (900, 600) 500	4	(630, 110), (760, 90), (710, 210), (610, 220) 255
5	(300, 500), (400, 600), (500, 600), (550, 500) 470	6	(110, 420), (210, 420), (210, 530), (110, 530) 420
7	(600, 700), (700, 700), (700, 600), (600, 500) 500	8	(420, 420), (470, 470), (530, 460), (490, 390), (440, 390) 600
9	(730, 740), (780, 740), (840, 790), (840, 840), (780, 890), (730, 840) 580	10	(210, 740), (260, 710), (320, 740), (340, 790), (260, 840), (190, 790) 400
11	(740, 420), (790, 500), (890, 530), (860, 470), (790, 420) 550		

Case 2

No.	Projection coordinates and height of obstacles	No.	Projection coordinates and height of obstacles
1	(120, 180), (180, 120), (240, 180) 350	2	(300, 140), (360, 200), (330, 260), (270, 220) 400
3	(116.7, 424.2), (220.3, 396.9), (276.6, 516.8), (180.4, 534.1) 550	4	(480, 110), (560, 130), (540, 220), (460, 210) 450
5	(650, 150), (720, 180), (740, 260), (680, 300), (620, 260) 500	6	(850, 110), (900, 160), (860, 220), (800, 200) 500
7	(420, 350), (470, 410), (530, 400), (490, 330), (440, 320) 650	8	(300, 600), (380, 640), (420, 700), (340, 720), (280, 670) 550
9	(580, 550), (660, 580), (750, 640), (640, 690), (560, 660) 600	10	(730, 450), (790, 500), (880, 520), (860, 460), (800, 430) 550
11	(110, 730), (170, 710), (240, 740), (260, 790), (200, 820) 320	12	(310, 760), (360, 780), (430, 810), (410, 860), (340, 830) 650
13	(510, 780), (560, 830), (630, 820), (590, 760) 700	14	(720, 760), (770, 780), (840, 830), (820, 880), (750, 850) 700
15	(580, 500), (690, 500), (690, 400), (650, 400), (650, 350), (610, 350), (610, 400), (580, 400) 600	16	(360, 580), (420, 620), (450, 500), (410, 520), (400, 500), (360, 520), (390, 550) 660
17	(890, 720), (930, 760), (960, 810), (910, 850), (860, 800) 700		

Case 3

No.	Projection coordinates and height of obstacles	No.	Projection coordinates and height of obstacles
1	(600, 400), (700, 500), (800, 400) 500	2	(320, 90), (410, 160), (440, 80), (380, 50) 600
3	(420, 850), (670, 800), (430, 760), (500, 700), (600, 900) 700	4	(150, 350), (200, 380), (180, 420), (130, 410), (100, 390) 500
5	(500, 200), (550, 250), (650, 200), (580, 150), (520, 130), (550, 150) 500	6	(700, 300), (750, 350), (800, 300), (780, 250), (730, 230), (710, 270) 600
7	(350, 650), (400, 700), (450, 650), (430, 600), (380, 580), (360, 620) 500	8	(850, 400), (900, 450), (950, 400), (930, 350), (880, 330), (890, 380) 388
9	(370, 270), (420, 320), (470, 270), (450, 220), (400, 200), (380, 240) 600	10	(150, 790), (100, 710), (300, 790) 680
11	(80, 850), (160, 870), (210, 930), (190, 970), (120, 985), (60, 950), (50, 890) 600	12	(80, 480), (160, 495), (200, 550), (150, 600), (70, 580), (55, 530), (65, 495) 590
13	(620, 900), (680, 945), (740, 930), (750, 880), (710, 830), (650, 845), (625, 870) 520	14	(801.13, 176.55), (849.11, 199.93), (897.07, 199.93), (945.05, 176.55), (962.02, 128.37), (945.05, 80.19), (897.07, 32.01), (849.11, 32.01), (801.13, 80.19), (784.16, 128.37) 480
15	(450, 550), (500, 500), (550, 550), (600, 500), (550, 450), (450, 450), (400, 500) 700	16	(420, 320), (470, 370), (520, 320), (470, 270), (420, 320), (370, 370), (320, 320), (370, 270) 600
17	(800, 600), (850, 650), (900, 600), (850, 550), (800, 600), (750, 650), (700, 600), (750, 550) 320	18	(100, 100), (200, 70), (250, 200), (150, 230), (80, 150), (130, 130) 550
19	(800, 800), (850, 750), (950, 800), (900, 850), (830, 880), (880, 820) 250	20	(300, 700), (350, 730), (400, 700), (380, 650), (320, 640), (340, 680) 560
21	(600, 100), (650, 150), (700, 100), (680, 50), (630, 30), (610, 80) 780	22	(200, 600), (250, 630), (300, 600), (280, 370), (230, 360), (210, 390) 660

Case 4

No.	Projection coordinates and height of obstacles	No.	Projection coordinates and height of obstacles
1	(426.02, 890.38), (354.61, 904.38), (343.12, 953.63), (283.33, 944.65), (407.26, 772.30) 382	2	(903.4, 887.3), (831.8, 927.5), (803.7, 881.4), (790.2, 886.7), (781.1, 861.5), (850.5, 786.6), (855.4, 767.5) 555
3	(900.5, 230.9), (821.6, 218.6), (819.7, 59.6), (900, 101.2), (938.1, 98.0) 557	4	(581.9, 515.8), (510.3, 561.1), (489.4, 561.8), (443.4, 499.5), (439.0, 467.3), (461.0, 434.4), (469.3, 425.9), (554.6, 459.1) 778

5	(375.8, 302.0), (371.6, 316.0), (316.1, 370.4), (308.4, 388.3), (240.3, 239.1), (315.7, 215.1), (351.5, 220.7), (379.3, 292.6) 575	6	(730.9, 767.4), (639.5, 709.3), (654.6, 635.8), (656.7, 625.9), (700, 700), (673.1, 616.8), (760.9, 681.7) 684
7	(342.5, 739.3), (349.2, 759.4), (303.6, 771.3), (293.1, 764.7), (246.4, 744.8), (287.2, 644.2), (285.1, 632.6), (307.7, 620.7) 780	8	(768.7, 334.2), (764.0, 368.9), (724.7, 397.4), (716.7, 212.9), (735.0, 222.5), (795.5, 288.1) 651
9	(554.6, 230.4), (535.7, 246.1), (427.6, 176.4), (464.3, 136.3), (484.2, 143.4), (517.7, 143.4) 508	10	(238.4, 822.5), (191.1, 857.5), (157.1, 866.1), (142.3, 833.9), (104.0, 792.4), (116.5, 775.4), (210.1, 761.6), (229.6, 793.1) 824
11	(310.7, 160.7), (191.3, 194.3), (171.6, 148.4), (198.1, 141.2), (176.5, 134.0), (179.4, 129.7), (295.3, 91.7) 459	12	(251.3, 541.9), (238.3, 553.9), (144.0, 540.3), (132.8, 486.8), (251.5, 476.2), (251.8, 489.8), (276.8, 487.1) 530
13	(720, 950), (720, 850), (690, 850), (690, 920), (650, 920), (650, 850), (620, 850), (620, 950), (720, 950) 800	14	(700, 100), (700, 200), (770, 200), (770, 170), (730, 170), (730, 100), (700, 100) 500
15	(900, 300), (915, 330), (925, 360), (930, 390), (925, 420), (915, 450), (900, 480), (885, 450), (875, 420), (870, 390), (875, 360), (885, 330), (900, 300) 600	16	(764.23, 225.37), (741.26, 260.65), (699.76, 263.87), (678.89, 235.90), (680.00, 203.01), (705.61, 185.22), (738.39, 190.45), (758.66, 211.51) 670.34
17	(934.64, 783.33), (928.77, 803.59), (778.20, 660.42), (834.26, 637.56) 852.43	18	(707.20, 478.92), (726.41, 507.48), (719.86, 537.64), (693.20, 552.90), (667.02, 537.16), (673.93, 503.81) 861.35
19	(586.22, 821.62), (568.39, 837.86), (574.23, 855.92), (516.80, 848.54), (499.75, 811.70), (488.67, 786.63), (514.71, 741.11), (513.48, 733.50), (528.05, 738.38), (592.24, 768.01), (596.05,	20	(186. 26, 631. 33), (190.48, 656.66), (172.16, 668.88), (145.78, 663.50), (124.44, 644.13), (116.64, 620.92), (127.14, 599.55), (150.35, 588.83), (175.72, 597.45), (186.81, 616.32) 880
21	(618.94, 187.67), (610.14, 211.53), (573.30, 229.91), (553.86, 217.74), (561.51, 191.78), (586.75, 176.00) 900	22	(185, 284), (121, 367), (34, 326), (35, 241), (116, 215), (202, 346), (157, 397), (90, 377), (75, 312), (120, 259), (183, 284) 680
23	(694.03, 523.45), (649.81, 545.82), (639.80, 450.08) 703.89	24	(340,560), (440,400), (280,450) 801
25	(750, 475), (750, 575), (780, 575), (780, 535), (820, 535), (820, 575), (850, 575), (850, 475), (820, 475), (820, 515), (780, 515), (780, 475), (750, 475) 790	26	(458.19, 285.60), (481.05, 299.88), (490.63, 322.67), (486.36, 348.69), (471.94, 366.03), (449.08, 371.67), (427.47, 360.08), (417.12, 337.85) 747.53
27	(460, 660), (484.44, 651.52), (484.44, 668.48), (448.28, 688.28), (471.57, 699.57), (459.57, 711.57), (420, 700), (428.48, 724.44) (411.52, 724.44), (391.72, 688.28), (380.43, 711.57), (368.43, 699.57), (380, 660), (355.56, 668.48), (355.56, 651.52), (391.72, 631.72), (368.43, 620.43), (380.43, 608.43), (420, 620), (411.52, 595.56), (428.48, 595.56), (448.28, 631.72), (459.57, 608.43), (471.57, 620.43), (460, 660) 620	28	(649.27, 350.00), (622.31, 362.36), (636.86, 388.21), (607.78, 382.36), (604.36, 411.82), (584.27, 390.00), (564.18, 411.82), (560.76, 382.36), (531.68, 388.21), (546.23, 362.36), (519.27, 350.00), (546.23, 337.64), (531.68, 311.79), (560.76, 317.64), (564.18, 288.18), (584.27, 310.00), (604.36, 288.18), (607.78, 317.64), (636.86, 311.79), (622.31, 337.64) 850
29	(370, 80), (420, 100), (465, 90), (515, 105), (575, 90), (610, 75), (600, 50), (555, 38), (505, 42), (455, 32), (410, 46), (385, 72) 280	30	(508.41, 897.14), (486.50, 924.52), (508.41, 951.91), (530.32, 930.00), (552.23, 951.91), (574.14, 924.52) 550

Case 5

No.	Projection coordinates and height of obstacles	No.	Projection coordinates and height of obstacles
1	(712.65, 169.16), (712.65, 269.16), (762.65, 269.16), (762.65, 219.16), (812.65, 219.16), (812.65, 269.16), (862.65, 269.16), (862.65, 169.16) 754.42	2	(571.78, 803.96), (565.84, 830.00), (549.19, 850.87), (525.13, 862.46), (498.43, 862.46), (474.37, 850.87), (457.72, 830.00), (451.78, 803.96) 448.07
3	(276.85, 305.15), (267.31, 330.24), (246.46, 346.67), (222.13, 351.95), (197.20, 345.17), (176.91, 327.27), (167.89, 302.31), (171.20, 277.22), (186.17, 257.15), (209.91, 247.09), (234.80, 248.27), (256.90, 260.17) 552.16	4	(173.52, 874.28), (217.85, 898.16), (267.34, 918.43),(234.12, 846.75), (241.76, 792.61), (193.25, 812.94), (156.43, 839.67), (173.52, 874.28) 500
5	(773.22, 424.33), (812.45, 468.28), (794.61, 509.13), (744.80, 518.45), (710.73, 487.22), (715.54, 439.68) 698.74	6	(628.77, 664.71), (661.34, 692.80), (696.23, 668.53), (684.58, 631.40), (647.36, 625.31), (623.09, 648.85) 893.99
7	(143.31, 692.25), (180.20, 740.59), (225.39, 775.37), (258, 755.72), (268.42, 707.90), (240.14, 664.38), (193.64, 662.56) 779.77	8	(571.90, 443.52), (627.87, 455.74), (640.15, 405.23), (582.47, 390.41) 631.66
9	(654.73, 523.15), (619.84, 502.61), (549.52, 557.03)·612.95	10	(343.71, 166.83), (459.60, 167.39), (444.99, 49.40) 473.55
11	(903.09, 371.99), (896.80, 391.29), (882.97, 404.89), (865.58, 410.76), (847.60, 412.01), (826.50, 409.91), (814.64, 392.26), (811.24, 371.99), (820.84, 354.71), (828.90, 337.09), (845.75, 323.86), (866.69, 328.38), (884.18, 337.58), (898.46, 351.90) 682.13	12	(905.58, 749.10), (903.55, 763.19), (897.64, 776.14), (888.32, 786.89), (876.35, 794.59), (862.69, 798.60), (848.46, 798.60), (834.81, 794.59), (822.83, 786.89), (813.51, 776.14), (807.60, 763.19), (805.58, 749.10) 862.37

13	(400. 97, 564.70), (393.72, 579.85), (388.41, 595.97), (373.97, 605.35), (357.13, 611.15), (339.90, 606.30), (328.00, 593.83), (313.82, 582.63), (316.24, 564.70), (318.99, 548.90), (327.53, 535.10), (341.69, 527.42), (357.13, 524.79), (372.33,528.02), (388.47, 533.36), (401.91, 546.15) 508.14	14	(514.97,214.14), (516.99, 200.06), (522.90, 187.11), (532.22, 176.36), (544.20, 168.66), (557.85, 164.65), (572.08, 164.65), (585.74, 168.66), (597.71, 176.36), (607.03, 187.11), (612.94, 200.06), (614.97, 214.14) 726.66
15	(500, 200), (550, 250), (650, 200), (580, 150), (520, 130), (550, 150) 500	16	(838.55, 715.38), (886.78, 732.11), (882.36, 771.89), (834.49, 763.22), (826.32, 739.60) 620.12
17	(478.86, 330.25), (492.05,355.09), (462.13, 350.18), (442.07, 371.91), (437.09, 342.10), (410.82, 329.01), (436.63, 314.04), (443.22, 284.55), (463.60, 306.21), (491.58, 299.31) 701.79	18	(150.11, 437.91), (190.27, 453.71), (198.62, 415.91), (159.64, 404.44) 604.36
19	(745.98, 289.62), (779.37, 317.58), (765.26, 350.67), (728.13, 350.67), (713.86, 317.74) 613.87	20	(714.27, 138.28), (694.27, 172.93), (654.27, 172.93), (634.27, 138.28), (654.27, 103.64), (694.27, 103.64) 823.94
21	(666.42, 738.33), (703.45, 764.61), (691.08, 804.18), (649.10, 804.18), (636.72, 764.61) 768.84	22	(345.10, 754.42), (375.91, 781.32), (406.74, 754.42), (375.91, 727.52) 559.41
23	(201.53, 621.91), (250.89, 621.91), (270.23, 586.11), (182.20, 586.11) 633.58	24	(479.38, 419.71), (531.42, 452.80), (502.27, 489.17), (450.82, 466.60) 625.84
25	(457.57, 270.89), (330.04, 292.49), (300.21, 270.89), (300.21, 234.89), (330.04, 213.29), (357.57, 234.89) 387.90	26	(300, 700), (350, 730), (400, 700), (380, 650), (320, 640), (340, 680) 700
27	(864.84, 585.46), (900.00, 550.30), (935.16, 585.46), (970.32, 550.30), (935.16, 515.14), (864.84, 515.14), (829.68, 550.30) 670	28	(540.10, 145.88), (570.88, 172.76), (610.54, 160.43), (579.76, 133.55) 644.21
29	(503.29, 563.07), (516.69, 594.28), (524.79, 625.49), (525.89, 656.70), (519.99, 687.91), (507.09, 719.12), (487.19, 750.33), (461.29, 719.12), (439.39, 687.91), (431.49, 656.70), (439.59, 625.49), (461.69, 594.28), (503.29, 563.07) 600	30	(804.63, 615.40), (798.52, 634.31), (784.50, 650.28), (764.39, 660.08), (741.17, 662.49), (718.47, 656.18), (700.37, 641.52), (690.91, 621.34), (690.91, 599.46), (700.37, 579.27), (718.47, 564.62), (741.17, 558.30) 659.55
31	(913.60, 221.71), (970.08, 221.71), (955.08, 206.71), (998.56, 206.71), (998.56, 163.23), (955, 163.23), (940.08, 148.23) 288.39	32	(800, 900), (850, 950), (900, 900), (850, 850), (800, 900), (750, 950), (700, 900), (750, 850) 600
33	(550, 300), (550, 320), (580, 320), (580, 370), (550, 370), (550, 390), (650, 390), (650, 370), (620, 370), (620, 320), (650, 320), (650, 300), (550, 300) 580	34	(225.52, 507.30), (225.52, 407.30), (275.52, 407.30), (275.52, 457.30), (325.52, 457.30), (325.52, 407.30), (375.52, 407.30), (375.52, 507.30) 859.16
35	(96.19, 141.31), (131.82, 147.28), (158.10, 111.18), (208.79, 134.24), (256.22, 104.61), (243.03, 74.61), (187.07, 89.95), (142.63, 104.15), (96.19, 141.31) 200	36	(374.73, 977.35), (518.86, 966.94), (507.65, 875.66), (477.22, 916.49), (468.42, 872.46), (444.39, 914.89), (438.79, 878.86), (418.77, 910.89), (401.16, 876.46) 650
37	(51.08, 545.37), (54.12, 526.23), (62.79, 510.01), (75.77, 499.17), (91.08, 495.37), (106.38, 499.17), (119.36, 510.01), (128.03, 526.23), (131.08, 545.37), (131.08, 695.37), (51.06, 695.37), (51.08, 545.37) 600	38	(227.46, 199.64), (281.61, 199.64), (301.67, 184.51), (281.61, 179.48), (301.67, 169.43), (259.58, 174.46), (254.55, 159.40), (248.53, 179.48) 300
39	(109.51, 216.79), (98.55, 229.93), (87.60, 244.17), (89.79, 262.80), (104.03, 277.04), (114.98, 282.51), (125.94, 282.51), (140.18, 273.75), (144.56, 258.41), (136.89, 244.17), (122.65, 229.93), (114.98, 222.26) 400	40	(732.19, 57.37), (732.19, 68.32), (825.30, 68.32), (836.26, 79.28), (876.79, 79.28), (883.36, 66.13), (876.79, 52.99), (836.26,52.99), (825.30, 57.37) 240

References

[1] W. Liu, T. Zhang, S. Huang, K. Li, A hybrid optimization framework for UAV reconnaissance mission planning, Computers & Industrial Engineering 173 (2022) 108653. https://doi.org/10.1016/j.cie.2022.108653.

[2] S. Asadzadeh, W.J.D. Oliveira, C.R.D. Souza Filho, UAV-based remote sensing for the petroleum industry and environmental monitoring: State-of-the-art and perspectives, Journal of Petroleum Science and Engineering 208 (2022) 109633. https://doi.org/10.1016/j.petrol.2021.109633.

[3] M. Niyazi, J. Behnamian, Application of Emerging Digital Technologies in Disaster Relief Operations: A Systematic Review, Arch Computat Methods Eng 30 (2023) 1579–1599. https://doi.org/10.1007/s11831-022-09835-3.

[4] F. Betti Sorbelli, UAV-Based Delivery Systems: A Systematic Review, Current Trends, and Research Challenges, ACM J. Auton. Transport. Syst. 1 (2024) 12:1-12:40. https://doi.org/10.1145/3649224.

[5] A.A. Laghari, A.K. Jumani, R.A. Laghari, H. Li, S. Karim, A.A. Khan, Unmanned aerial vehicles advances in object detection and communication security review, Cognitive Robotics 4 (2024) 128–141. https://doi.org/10.1016/j.cogr.2024.07.002.

[6] A.A. Laghari, A.K. Jumani, R.A. Laghari, H. Nawaz, Unmanned aerial vehicles: A review, Cognitive Robotics 3 (2023) 8–22. https://doi.org/10.1016/j.cogr.2022.12.004.

[7] A.A. Khan, A.A. Laghari, T.R. Gadekallu, Z.A. Shaikh, A.R. Javed, M. Rashid, V.V. Estrela, A. Mikhaylov, A drone-based data management and optimization using metaheuristic algorithms and blockchain smart contracts in a secure fog environment, Computers and Electrical Engineering 102 (2022) 108234. https://doi.org/10.1016/j.compeleceng.2022.108234.

[8] X. Chai, Z. Zheng, J. Xiao, L. Yan, B. Qu, P. Wen, H. Wang, Y. Zhou, H. Sun, Multi-strategy fusion differential evolution algorithm for UAV path planning in complex environment, Aerospace Science and Technology 121 (2022) 107287. https://doi.org/10.1016/j.ast.2021.107287.

[9] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, L. Jurišica, Path Planning with Modified a Star Algorithm for a Mobile Robot, Procedia Engineering 96 (2014) 59–69. https://doi.org/10.1016/j.proeng.2014.12.098.

[10] B. Li, B. Chen, An Adaptive Rapidly-Exploring Random Tree, IEEE/CAA Journal of Automatica Sinica 9 (2022) 283–294. https://doi.org/10.1109/JAS.2021.1004252.

[11] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE Comput. Intell. Mag. 1 (2006) 28–39. https://doi.org/10.1109/MCI.2006.329691.

[12] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, IEEE, Perth, WA, Australia, 1995: pp. 1942–1948. https://doi.org/10.1109/ICNN.1995.488968.

[13] J.H. Holland, Genetic Algorithms, Scientific American 267 (1992) 66–73.
[14] S. Das, P.N. Suganthan, Differential Evolution: A Survey of the State-of-the-Art, IEEE Trans. Evol. Computat. 15 (2011) 4–31. https://doi.org/10.1109/TEVC.2010.2059031.
[15] X. Yu, C. Li, J. Zhou, A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios, Knowledge-Based Systems 204 (2020) 106209. https://doi.org/10.1016/j.knosys.2020.106209.
[16] M.F. Ahmad, N.A.M. Isa, W.H. Lim, K.M. Ang, Differential evolution: A recent review based on state-of-the-art works, Alexandria Engineering Journal 61 (2022) 3831–3872. https://doi.org/10.1016/j.aej.2021.09.013.
[17] C. Zhu, Y. Bouteraa, M. Khishe, D. Martín, F. Hernando-Gallego, T. Vaiyapuri, Enhancing unmanned marine vehicle path planning: A fractal-enhanced chaotic grey wolf and differential evolution approach, Knowledge-Based Systems 317 (2025) 113481. https://doi.org/10.1016/j.knosys.2025.113481.
[18] A.A. Laghari, A. Khan, Hui He, V.V. Estrela, N. Razmjooy, J. Hemanth, H.J. Loschi, Quality of experience (QoE) and quality of service (QoS) in UAV systems, in: V.V. Estrela, J. Hemanth, O. Saotome, G. Nikolakopoulos, R. Sabatini (Eds.), Imaging and Sensing for Unmanned Aircraft Systems: Volume 2: Deployment and Applications, Institution of Engineering and Technology, 2020: pp. 215–245. https://doi.org/10.1049/PBCE120G_ch10.
[19] F. Ling, C. Du, J. Chen, Z. Yuan, An Improved Geometrical Path Planning Algorithm for UAV in Irregular-obstacle Environment, in: 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), IEEE, Chongqing, China, 2019: pp. 972–976. https://doi.org/10.1109/ITAIC.2019.8785442.
[20] Y. Zhang, W. Xiao, S. Yin, EAB-BES: A Global Optimization Approach for Efficient UAV Path Planning in High-Density Urban Environments, Biomimetics 10 (2025) 499. https://doi.org/10.3390/biomimetics10080499.
[21] M. Khishe, Greedy opposition-based learning for chimp optimization algorithm, Artif Intell Rev 56 (2023) 7633–7663. https://doi.org/10.1007/s10462-022-10343-w.
[22] N. Bashir, S. Boudjit, G. Dauphin, S. Zeadally, An obstacle avoidance approach for UAV path planning, Simulation Modelling Practice and Theory 129 (2023) 102815. https://doi.org/10.1016/j.simpat.2023.102815.
[23] A.H. Alsabhan, S. Malik, S.G. Krishna Patro, C. Mahanty, A.A. Hadi, M. Ghouse, A. Thiyagarajan, M. Mittal, M. Khishe, SailMutLoc: a sailfish-mutation enhanced optimization algorithm for student performance prediction, Ain Shams Engineering Journal 17 (2026) 103898. https://doi.org/10.1016/j.asej.2025.103898.
[24] T.M. Cabreira, P.R. Ferreira, C.D. Franco, G.C. Buttazzo, Grid-Based Coverage Path Planning With Minimum Energy Over Irregular-Shaped Areas With Uavs, in: 2019 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, Atlanta, GA, USA, 2019: pp. 758–767. https://doi.org/10.1109/ICUAS.2019.8797937.

[25] Y. Choi, Y. Choi, S. Briceno, D.N. Mavris, Three-dimensional UAS Trajectory Optimization for Remote Sensing in an Irregular Terrain Environment, in: 2018 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, Dallas, TX, 2018: pp. 1101–1108. https://doi.org/10.1109/ICUAS.2018.8453310.

[26] M. Jones, S. Djahel, K. Welsh, Path-Planning for Unmanned Aerial Vehicles with Environment Complexity Considerations: A Survey, ACM Comput. Surv. 55 (2023) 234:1-234:39. https://doi.org/10.1145/3570723.

[27] E. Deilam Salehi, M. Fazli, Late acceptance hill climbing based algorithm for Unmanned Aerial Vehicles (UAV) path planning problem, Applied Soft Computing 170 (2025) 112651. https://doi.org/10.1016/j.asoc.2024.112651.

[28] Jingqiao Zhang, A.C. Sanderson, JADE: Adaptive Differential Evolution With Optional External Archive, IEEE Trans. Evol. Computat. 13 (2009) 945–958. https://doi.org/10.1109/TEVC.2009.2014613.

[29] R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for Differential Evolution, in: 2013 IEEE Congress on Evolutionary Computation, IEEE, Cancun, Mexico, 2013: pp. 71–78. https://doi.org/10.1109/CEC.2013.6557555.

[30] R. Tanabe, A.S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, Beijing, China, 2014: pp. 1658–1665. https://doi.org/10.1109/CEC.2014.6900380.

[31] A.W. Mohamed, A.A. Hadi, A.M. Fattouh, K.M. Jambi, LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems, in: 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE, Donostia, San Sebastián, Spain, 2017: pp. 145–152. https://doi.org/10.1109/CEC.2017.7969307.

[32] L. Zhu, Y. Zhou, G. Zhou, Q. Luo, Y. Wei, Solving spherical multi-aircraft path planning problem using self-adaptive update strategy differential evolution algorithm, Swarm and Evolutionary Computation 96 (2025) 102004. https://doi.org/10.1016/j.swevo.2025.102004.

[33] R. Chen, B. Yang, S. Li, S. Wang, A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem, Computers & Industrial Engineering 149 (2020) 106778. https://doi.org/10.1016/j.cie.2020.106778.

[34] Y. Cao, H. Zhang, W. Li, M. Zhou, Y. Zhang, W.A. Chaovalitwongse, Comprehensive Learning Particle Swarm Optimization Algorithm With Local Search for Multimodal Functions, IEEE Trans. Evol. Computat. 23 (2019) 718–731. https://doi.org/10.1109/TEVC.2018.2885075.

[35] R. Shang, Y. Wang, J. Wang, L. Jiao, S. Wang, L. Qi, A multi-population cooperative coevolutionary algorithm for multi-objective capacitated arc routing problem, Information Sciences 277 (2014) 609–642. https://doi.org/10.1016/j.ins.2014.03.008.

[36] X. Li, L. Wang, Q. Jiang, N. Li, Differential evolution algorithm with multi-population cooperation and multi-strategy integration, Neurocomputing 421 (2021) 285–302.

https://doi.org/10.1016/j.neucom.2020.09.007.
[37] Y. Shen, J. Wu, M. Ma, X. Du, H. Wu, X. Fei, D. Niu, Improved differential evolution algorithm based on cooperative multi-population, Engineering Applications of Artificial Intelligence 133 (2024) 108149. https://doi.org/10.1016/j.engappai.2024.108149.
[38] Q. Wang, Y. Li, Z. Hou, J. Zou, J. Zheng, A novel multi-population evolutionary algorithm based on hybrid collaboration for constrained multi-objective optimization, Swarm and Evolutionary Computation 87 (2024) 101581. https://doi.org/10.1016/j.swevo.2024.101581.
[39] J. Zou, R. Sun, Y. Liu, Y. Hu, S. Yang, J. Zheng, K. Li, A Multipopulation Evolutionary Algorithm Using New Cooperative Mechanism for Solving Multiobjective Problems With Multiconstraint, IEEE Trans. Evol. Computat. 28 (2024) 267–280. https://doi.org/10.1109/TEVC.2023.3260306.
[40] L. Fu, H. Ouyang, C. Zhang, S. Li, A.W. Mohamed, A constrained cooperative adaptive multi-population differential evolutionary algorithm for economic load dispatch problems, Applied Soft Computing 121 (2022) 108719. https://doi.org/10.1016/j.asoc.2022.108719.
[41] Y. Lu, B. Li, S. Liu, A. Zhou, A Population Cooperation based Particle Swarm Optimization algorithm for large-scale multi-objective optimization, Swarm and Evolutionary Computation 83 (2023) 101377. https://doi.org/10.1016/j.swevo.2023.101377.
[42] X. Bai, H. Jiang, C. Li, I. Ullah, M.M.A. Dabel, A.K. Bashir, Z. Wu, S.S. Ge, Efficient Hybrid Multi-Population Genetic Algorithm for Multi-UAV Task Assignment in Consumer Electronics Applications, IEEE Trans. Consumer Electron. 71 (2025) 2395–2406. https://doi.org/10.1109/TCE.2025.3563339.
[43] W. Liu, R. Wang, T. Zhang, K. Li, W. Li, H. Ishibuchi, X. Liao, Hybridization of Evolutionary Algorithm and Deep Reinforcement Learning for Multiobjective Orienteering Optimization, IEEE Trans. Evol. Computat. 27 (2023) 1260–1274. https://doi.org/10.1109/TEVC.2022.3199045.
[44] F. Wang, X. Wang, S. Sun, A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization, Information Sciences 602 (2022) 298–312. https://doi.org/10.1016/j.ins.2022.04.053.
[45] Z. Hu, W. Gong, W. Pedrycz, Y. Li, Deep reinforcement learning assisted co-evolutionary differential evolution for constrained optimization, Swarm and Evolutionary Computation 83 (2023) 101387. https://doi.org/10.1016/j.swevo.2023.101387.
[46] A.D. Martinez, J. Del Ser, E. Osaba, F. Herrera, Adaptive Multifactorial Evolutionary Optimization for Multitask Reinforcement Learning, IEEE Trans. Evol. Computat. 26 (2022) 233–247. https://doi.org/10.1109/TEVC.2021.3083362.
[47] F. Song, H. Xing, X. Wang, S. Luo, P. Dai, Z. Xiao, B. Zhao, Evolutionary Multi-Objective Reinforcement Learning Based Trajectory Control and Task Offloading in UAV-Assisted Mobile Edge Computing, IEEE Trans. on Mobile Comput. (2022) 1–18.

https://doi.org/10.1109/TMC.2022.3208457.
[48] S.C. Choi, Y. Lee, S.W. Cho, Reinforcement learning-integrated evolutionary algorithm for enhanced unmanned aerial vehicle coverage path planning, Swarm and Evolutionary Computation 97 (2025) 102051. https://doi.org/10.1016/j.swevo.2025.102051.
[49] A. Hajijamali Arani, M.M. Azari, P. Hu, Y. Zhu, H. Yanikomeroglu, S. Safavi-Naeini, Reinforcement Learning for Energy-Efficient Trajectory Design of UAVs, IEEE Internet Things J. 9 (2022) 9060–9070. https://doi.org/10.1109/JIOT.2021.3118322.
[50] J. Chen, C. Zhao, Q. Wang, H. Meng, HMANet: Hyperbolic Manifold Aware Network for Skeleton-Based Action Recognition, IEEE Trans. Cogn. Dev. Syst. 15 (2023) 602–614. https://doi.org/10.1109/TCDS.2022.3171550.
[51] F. Cao, Z. Tang, C. Zhu, X. Zhao, Manifold-guided multi-objective gradient algorithm combined with adjoint method for supersonic aircraft shape design, Aerospace Science and Technology 147 (2024) 109063. https://doi.org/10.1016/j.ast.2024.109063.
[52] B. Zhou, F. Gao, J. Pan, S. Shen, Robust Real-time UAV Replanning Using Guided Gradient-based Optimization and Topological Paths, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Paris, France, 2020: pp. 1208–1214. https://doi.org/10.1109/ICRA40945.2020.9196996.
[53] A. Orthey, M. Toussaint, Section Patterns: Efficiently Solving Narrow Passage Problems in Multilevel Motion Planning, IEEE Trans. Robot. 37 (2021) 1891–1905. https://doi.org/10.1109/TRO.2021.3070975.
[54] W. Yao, B. Lin, B.D.O. Anderson, M. Cao, Topological Analysis of Vector-Field Guided Path Following on Manifolds, IEEE Trans. Automat. Contr. 68 (2023) 1353–1368. https://doi.org/10.1109/TAC.2022.3151236.
[55] M. Zhang, Y. Han, S. Chen, M. Liu, Z. He, N. Pan, A Multi-Strategy Improved Differential Evolution algorithm for UAV 3D trajectory planning in complex mountainous environments, Engineering Applications of Artificial Intelligence 125 (2023) 106672. https://doi.org/10.1016/j.engappai.2023.106672.
[56] B. Shen, M. Khishe, S. Mirjalili, Evolving Marine Predators Algorithm by dynamic foraging strategy for real-world engineering optimization problems, Engineering Applications of Artificial Intelligence 123 (2023) 106207. https://doi.org/10.1016/j.engappai.2023.106207.
[57] Y. Zhang, Y. Zhou, S. Chen, W. Xiao, M. Wu, Bald eagle search algorithm for solving a three-dimensional path planning problem, MBE 21 (2024) 2856–2878. https://doi.org/10.3934/mbe.2024127.
[58] J. Yang, F. Yan, J. Zhang, C. Peng, Hybrid chaos game and grey wolf optimization algorithms for UAV path planning, Applied Mathematical Modelling 142 (2025) 115979. https://doi.org/10.1016/j.apm.2025.115979.
[59] F. Cui, Z.A. Al-Sudani, G.S. Hassan, H.A. Afan, S.J. Ahammed, Z.M. Yaseen, Boosted artificial intelligence model using improved alpha-guided grey wolf optimizer for

groundwater level prediction: Comparative study and insight for federated learning technology, Journal of Hydrology 606 (2022) 127384. https://doi.org/10.1016/j.jhydrol.2021.127384.
[60] D. Zhu, S. Wang, C. Zhou, S. Yan, J. Xue, Human memory optimization algorithm: A memory-inspired optimizer for global optimization problems, Expert Systems with Applications 237 (2024) 121597. https://doi.org/10.1016/j.eswa.2023.121597.
[61] X. Xu, C. Xie, Z. Luo, C. Zhang, T. Zhang, A multi-objective evolutionary algorithm based on dimension exploration and discrepancy evolution for UAV path planning problem, Information Sciences 657 (2024) 119977. https://doi.org/10.1016/j.ins.2023.119977.
[62] Z. Bai, H. Zhou, J. Wei, X. Zhou, Y. Ning, J. Wang, An exploration-enhanced hybrid algorithm based on regularity evolution for multi-objective multi-UAV 3-D path planning, Complex Intell. Syst. 11 (2025) 225. https://doi.org/10.1007/s40747-025-01846-4.
[63] P. Jangir, Arpita, S.B. Pandya, G. G, M. Khishe, B.I. Trivedi, MaOAOA: A Novel Many-Objective Arithmetic Optimization Algorithm for Solving Engineering Problems, Engineering Reports 7 (2025) e70077. https://doi.org/10.1002/eng2.70077.
[64] Q. Wang, G. Chen, M. Khishe, B.F. Ibrahim, S. Rashidi, Multi-objective optimization of IoT-based green building energy system using binary metaheuristic algorithms, Journal of Building Engineering 68 (2023) 106031. https://doi.org/10.1016/j.jobe.2023.106031.
[65] E. Yanmaz, H.M. Balanji, İ. Güven, Dynamic Multi-UAV Path Planning for Multi-Target Search and Connectivity, IEEE Trans. Veh. Technol. 73 (2024) 10516–10528. https://doi.org/10.1109/TVT.2024.3363840.
[66] S. Yin, Z. Xiang, Real-time distributed decision-making for simultaneous target assignment and path planning in multiple unmanned surface vehicles, Expert Systems with Applications 279 (2025) 127457. https://doi.org/10.1016/j.eswa.2025.127457.
[67] A.A. Khan, A.A. Laghari, M. Shafiq, S.A. Awan, Z. Gu, Vehicle to Everything (V2X) and Edge Computing: A Secure Lifecycle for UAV-Assisted Vehicle Network and Offloading with Blockchain, Drones 6 (2022) 377. https://doi.org/10.3390/drones6120377.
[68] S. Yin, H. Li, A.A. Laghari, T.R. Gadekallu, G.A. Sampedro, A. Almadhor, An Anomaly Detection Model Based on Deep Auto-Encoder and Capsule Graph Convolution via Sparrow Search Algorithm in 6G Internet of Everything, IEEE Internet of Things Journal 11 (2024) 29402–29411. https://doi.org/10.1109/JIOT.2024.3353337.