I.J. Information Technology and Computer Science, **2025, 6, 109-126**
Published Online on December 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2025.06.06

A Novel Hybrid Differential Evolution and Enhanced Whale Optimization Algorithm for UAV Path Planning

Mykola Nikolaiev
Department of computer engineering, National Technical University of Ukraine, "Igor Sikorsky Kyiv Polytechnic Institute" Kyiv, 03056, Ukraine
E-mail: nickolay.dev@gmail.com
ORCID iD: https://orcid.org/0009-0003-0787-6726

Mykhailo Novotarskyi
Department of computer engineering, National Technical University of Ukraine, "Igor Sikorsky Kyiv Polytechnic Institute" Kyiv, 03056, Ukraine
E-mail: novotar@gmail.com
ORCID iD: https://orcid.org/0000-0002-5653-8518

Artem Volokyta*
Department of computer engineering, National Technical University of Ukraine, "Igor Sikorsky Kyiv Polytechnic Institute"/ Kyiv, 03056, Ukraine
E-mail: artem.volokita@kpi.ua
ORCID iD: https://orcid.org/0000-0001-9069-5544
*Corresponding author

Received: 24 June 2025; Revised: 06 September 2025; Accepted: 02 October 2025; Published: 08 December 2025

Abstract: Safe and energy-aware navigation for unmanned aerial vehicles (UAVs) requires the simultaneous optimization of path length, curvature, obstacle clearance, altitude, energy expenditure, and mission time—within the tight computational limits of on-board processors. This study proposes a two-phase hybrid optimizer that couples the global search capability of Differential Evolution (DE) with an Enhanced Whale Optimization Algorithm (E-WOA) specialized for local refinement. E-WOA improves on the canonical WOA through three principled modifications: real-time boundary repair to ensure path feasibility, quasi-oppositional learning to restore population diversity, and an adaptive stagnation trigger that re-initiates exploration when progress stalls. When the population's improvement plateaus, control transfers from DE to E-WOA, combining broad exploration with focused exploitation. Comparative experiments conducted in 3D environments with static obstacles that block direct line-of-sight routes demonstrate that the hybrid achieves lower composite cost—normalized over path length, curvature, risk, altitude, energy and time—shorter and smoother trajectories, and faster convergence than standard metaheuristics while preserving obstacle clearances and curvature limits. Averaged over 30 independent trials, our hybrid framework reduced the normalized composite cost by 14.5% relative to the next-best algorithm (Grey Wolf Optimizer) and produced feasible paths in an average of 2.35 seconds on commodity hardware—adequate for strategic re-planning, though further optimization is needed for sub-second control loops. Blending DE's global reach with a diversity-aware, adaptively stalled WOA provides a practical foundation for strategic, near-real-time replanning in 3D airspaces.

Index Terms: UAV, Path Planning, Differential Evolution, Whale Optimization Algorithm, Evolutionary Computation, Autonomous Flight Optimization, Optimization.

1. Introduction

Unmanned Aerial Vehicles (UAVs) have evolved to fulfill critical roles across various sectors, including military reconnaissance, disaster response, precision agriculture, environmental inspection, and automated logistics management [1]. Because UAV applications are expanding rapidly, advanced path-planning strategies are essential to ensure safe, efficient, and versatile navigation [2]. UAV path planning must jointly satisfy obstacle avoidance, flight-stability, power-efficiency, and computational-latency constraints, making it an inherently sophisticated multi-objective problem [3,4].

Classical graph-based and potential-field planners struggle in high-dimensional, uncertain, or dynamic flight scenarios [5,6]. They tend to converge to local optima, demand high computational resources, and cannot support real-time decision making. Evolutionary algorithms are attractive for their ability to explore large search spaces, adapt to environmental changes, and deliver near-optimal solutions within practical time limits [7,8]. Recent work shows that hybrid metaheuristics handle complex optimisation landscapes more effectively than single-strategy methods [9].

However, a practical path planning solution must extend beyond mere optimization to address the physical realities of UAV operation. These include kinodynamic constraints, such as maximum turn rates and acceleration, which preclude instantaneous changes in trajectory. Operational factors like limited battery endurance demand energy-efficient paths, while external variables such as wind strongly affect stability and energy use. Furthermore, missions are often subject to regulatory constraints, including geofenced no-fly zones. Although we do not model full UAV dynamics or environmental stochasticity, our framework encodes these operational constraints in a multi-objective cost function. Our approach seeks to generate paths that are short, collision-free, smoother, energy-efficient, and compliant with defined operational boundaries, rendering them suitable as reference trajectories for a UAV's low-level flight controller.

Differential Evolution (DE) is valued for its strong global-search capability in high-dimensional spaces [10]. However, despite these advantages, DE is prone to premature convergence and has limited local exploitation capabilities, often resulting in suboptimal solutions in complex environments [11]. Similarly, the Whale Optimization Algorithm (WOA), inspired by humpback whales' bubble-net hunting behavior, offers effective local search but is prone to local-minimum traps and slow convergence on intricate problems [12]. To address these limitations, our study introduces a hybrid optimization framework that integrates the robust global exploration capabilities of DE with the enhanced local exploitation features of the Enhanced WOA variant (E-WOA). Despite recent advances, existing metaheuristics have demonstrated limited effectiveness in terms of rapid convergence and local exploitation when applied individually. This study addresses this research gap by combining the strengths of DE (global exploration) and E-WOA (local exploitation), aiming at optimized performance in dynamic and obstacle-rich UAV path planning. The Enhanced WOA (E-WOA) proposed here incorporates adaptive parameter tuning, quasi-oppositional learning, and hybrid mutation strategies, distinctively improving local exploitation and avoiding local optima. This combined DE + E-WOA approach leverages the complementary strengths of both algorithms, promoting balanced exploration and exploitation to achieve improved performance in UAV path planning tasks. Specifically, the hybrid framework seeks to accelerate convergence, enhance path smoothness, and effectively avoid obstacles, thereby addressing the limitations of conventional methods in dynamic, three-dimensional navigation environments.

The remainder of this paper is structured as follows. Section 2 provides a comprehensive literature review of both traditional and metaheuristic UAV path planning techniques, establishing the context for our contribution. Section 3 formalizes the path-planning problem, including the mathematical formulation of the multi-objective cost function used throughout the study. Section 4 details the proposed hybrid methodology, explaining the individual roles of Differential Evolution and the Enhanced Whale Optimization Algorithm, as well as their synergistic integration. Section 5 describes the experimental setup and presents a comparative analysis of the simulation results. Finally, Section 6 concludes with a synthesis of the principal results, acknowledging the study's limitations, and outlining promising directions for future research.

2. Literature Review

UAV systems require sophisticated optimizers to generate trajectories that maximize safety and efficiency [13]. In current practice, UAV path planning often involves modeling the flight environment mathematically to formulate the path planning problem as an optimization task [14], which can then be solved using various algorithms. This enables the transformation of environmental constraints into a framework that supports the development of safe, feasible, and stable flight paths from the starting point to the destination [15,16]. Key objectives in UAV path planning include minimizing flight time, avoiding obstacles, reducing power consumption, and maintaining aerodynamic stability. The increasing complexity of modern UAV operations renders traditional path-planning strategies ineffective against real-world challenges, such as unpredictable obstacles, uncertain terrain, and high-dimensional environments [17].

Metaheuristic methods are vital for UAV path planning, enabling efficient exploration of large solution spaces with dynamic adaptability and robust management of multiple performance criteria [18]. The algorithms serve their purpose especially in situations that demand extensive search methods cannot run efficiently. Consequently, recent research focuses on hybridizing evolutionary and swarm-intelligence algorithms for complex UAV navigation.

Traditional UAV path-planning techniques use three primary methods which include graph-based methods together with potential field algorithms and rule-based heuristics to find optimal flight paths [19]. Despite their intuitive appeal, potential field methods are susceptible to local minima, causing UAV trajectories to become trapped and inefficient in obstacle-dense scenarios, severely limiting their practical utility in complex, cluttered environments. Current navigation methods use pre-established mathematical equations with defined search methods to establish optimal flight routes. When applied to real-time UAV operations these methods face various limitations even though they possess strong theoretical capabilities.

In graph-based path planning algorithms, the flight environment is modeled as a discrete network, where each possible position of the UAV is represented as a node, and the feasible movements between these positions are denoted by edges connecting the nodes [20]. Algorithms such as Dijkstra's Algorithm and A* [21,22] are employed to determine the optimal path from a starting node to a goal node by evaluating cost functions based on metrics like Euclidean distance or time expenditure. This approach allows for systematic exploration of the network to identify paths that minimize the specified cost, making it a fundamental technique in UAV navigation and trajectory optimization. Despite their effectiveness in finding optimal paths, these algorithms exhibit notable limitations in terms of computational efficiency. The computational complexity of Dijkstra's Algorithm results in considerable processing time, rendering it impractical for real-time applications, particularly in scenarios involving large graphs. This slowness stems from its exhaustive search nature, which evaluates all possible paths, making it less suitable for time-sensitive or extensive UAV missions. A*, an advanced derivative of Dijkstra's Algorithm, incorporates heuristic information to enhance performance, significantly reducing computation time compared to its predecessor. However, despite this improvement, A* still demands substantial computational resources when operating in environments that are complex, densely populated with obstacles, or subject to dynamic changes, limiting its scalability and adaptability in challenging operational contexts. The effectiveness of graph-based techniques decreases in dynamic, unpredictable environments with emerging obstacles [20]. The use of discretized space representations and their computational demands increases the complexity of establishing real-time UAV navigation systems. Another traditional method is the Rapidly Exploring Random Tree (RRT) algorithm [23,24], a sampling-based approach that avoids discretizing the flight environment, enabling faster search times. However, RRT often struggles to produce optimal trajectories, generating paths that are feasible but not necessarily efficient. Additionally, in large-scale and high-dimensional spaces, potential field methods [25,26] may experience local oscillations and fail to generate practical paths, while A*—despite its heuristic efficiency—demands substantial computational resources, limiting its effectiveness in complex or dynamic operational contexts.

Potential field methods are a class of algorithms, utilizing physics-based models to direct the UAV's movement. In this approach, the UAV is conceptualized as a charged particle influenced by two primary types of forces: attractive forces, which pull the UAV towards its designated destination, and repulsive forces, which push it away from potential obstacles. This framework provides an intuitive and easily implementable strategy for trajectory design, as it emulates natural physical interactions to guide the UAV through its environment.

Despite their appeal and simplicity, potential field methods have notable limitations. One critical drawback is localization, a situation where the attractive and repulsive forces acting on the UAV reach a balance, creating an immobilization point or local minimum that traps the UAV and hinders its progress toward the target. In obstacle-dense environments, the method often exhibits erratic behavior, failing to produce efficient, collision-free paths. Instead, the UAV may follow erratic and inefficient routes as it struggles to reconcile the competing forces. These shortcomings motivate the search for more robust methods, especially in cluttered environments.

A key limitation of these classical planners is their inherent difficulty in natively handling complex, multi-objective optimization problems where criteria such as path smoothness, energy consumption, and risk must be balanced simultaneously. They are typically designed to optimize a single metric, such as path length. Metaheuristic algorithms, in contrast, are exceptionally well-suited for this task. They provide a flexible framework to navigate the complex trade-offs between multiple competing objectives, making them a powerful tool for finding high-quality, practical solutions in the multi-faceted problem space of modern UAV missions [27].

UAV path-planning problems have been successfully solved with metaheuristic algorithms which can explore vast search spaces and adapt to changing constraints [28]. Intelligent algorithms for UAV path planning can be classified into three categories: traditional optimization methods, intelligent optimization techniques, and machine learning approaches. Metaheuristic methods, a subset of intelligent optimization algorithms inspired by natural phenomena, have become essential due to their ability to efficiently search large solution spaces and adapt to dynamic constraints. Beyond the commonly used Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), DE, and WOA, other examples include the Social Spider Algorithm (SSA) [29], Grey Wolf Optimization (GWO) [30], Beluga Whale Optimization (BWO) [31,32], Sine Cosine Algorithm (SCA) [33], and Butterfly Optimization Algorithm (BOA) [34-36], which employ diverse search strategies to identify optimal paths. However, PSO and GWO often converge prematurely in high-dimensional spaces, limiting their practical use.

- The Genetic Algorithms (GA) apply evolutionary characteristics from natural selection to guide a population of candidate solutions. Through selection crossover and mutation operations this algorithm produces high-quality solutions with each iterative process. GA successfully plans UAV paths however its performance slows down during execution of complex pathfinding tasks.
- PSO stands as an optimization technique based on bird and fish social behavior which engineers use regularly for UAV trajectory optimization [37]. The method delivers rapid optimization together with easy execution though it experiences early termination issues during pathfinding operations in complicated environments.
- ACO follows the ant colony foraging patterns to develop optimum UAV routes through route reinforcement in iterative processes. This optimization approach works well in grid structures while big-scale path planning problems lead to problems with stagnation.
- DE functions as a robust global optimization technique which demonstrates powerful capabilities especially when searching areas of high dimensions. The optimization system uses crossover combined with differential

mutation to both sustain population dispersal and identify efficient optimal paths. The value of DE as an optimization method can be improved through collaborations with other artificial intelligence techniques.

- The WOA draws its inspiration from humpback whale hunting practices to deliver refined solutions through its three move sets which include encircling and bubble-net attacking and spiral movements [38]. WOA delivers improved performance for UAV path planning through its ability to enhance both solution excellence and adjusting properties. Several enhancements to the WOA have been proposed by researchers to further enhance its effectiveness in UAV path planning. For example, the whale army optimization algorithm adjusts key parameters to achieve faster convergence with lower computational complexity [39]. Other improvements include segmented learning with adaptive operator selection [40], wavelet mutation strategies and social learning mechanisms [41], which enhance global exploration and help avoid local optima. These advancements reflect ongoing efforts to refine metaheuristic algorithms, making them more effective for UAV navigation in complex environments.

Moreover, driven by rapid advancements in machine learning, researchers are increasingly investigating its potential for UAV path planning. Techniques such as reinforcement learning and neural networks provide opportunities to derive optimal path-planning strategies from data, enabling adaptation to complex and dynamic environments. While still an emerging field, this represents a promising direction for future research in UAV navigation, complementing the capabilities of metaheuristic approaches.

Recently, research has shifted toward hybrids that combine evolutionary and swarm-intelligence methods to address UAVs' multi-objective demands. Systematic syntheses [42,43] have surveyed dozens of studies and concluded that hybrids outperform single-paradigm methods across nearly every key metric—trajectory feasibility, energy efficiency, and computational throughput—particularly when obstacle fields are dense or mission objectives conflict.

Against this backdrop a second, more nuanced wave of research has focused on tailoring exploration–exploitation splits to specific operational demands. Adam et al. fused Artificial Bee Colony search with GWO to coordinate multi-UAV disaster-response teams, obtaining resilient group trajectories despite communication loss [44]. Baidya et al. showed that a PSO-ABC pairing can prolong battery life under dynamic threat levels [45], whereas Altun & Aydın demonstrated that PSO-GWO hybrids smooth long-range flight paths more effectively than either constituent alone [46]. Similar division-of-labor rationales appear in the Water Cycle Algorithm–Harmony Search hybrid for indoor micro-UAVs [47], the multi-strategy Ant Colony variant MAHACO for swarm-scale 3D missions [48], and Cuckoo-based sweep-coverage planner that combines global area allocation with local refinement [49].

Concurrently, algorithmic diversity has broadened. Shen et al. appended reflection rules, Lévy flights, and crossover operations to a Dung Beetle Optimizer [50], while Belge et al. coupled Harris Hawk Optimization with GWO to respect payload-induced energy budgets [51]. Immune-inspired and adaptive frameworks—DUALIPA [52] and the adaptive Dung Beetle scheme [53]—have pushed the envelope further by embedding real-time parameter control and energy-aware cost modelling into the optimization loop. Collectively, the literature shows that no single metaheuristic delivers global reach, local refinement, and low latency simultaneously; well-designed hybrids are therefore essential.

Despite these advances, two persistent gaps remain. First, many hybrids still suffer from either weak local exploitation or premature stagnation once population diversity collapses (e.g., pure WOA, PSO, or ACO variants). Second, only a handful of studies explicitly embed robustness mechanisms—such as boundary repair, diversity injection, or stagnation detectors—into the algorithm's core dynamics, leaving performance brittle in highly cluttered or rapidly changing environments.

Our DE + E-WOA framework directly addresses these shortcomings. Building on the exploration strength of DE, we graft an E-WOA exploitation phase that clamps infeasible solutions in real time to avoid wasted evaluations, injects quasi-oppositional candidates to preserve diversity, and activates an adaptive stagnation-response trigger to reignite global search whenever progress stalls. The proposed two-stage scheme therefore unites the broad exploration of DE-based hybrids with the precise local refinement and adaptive safeguards of recent multi-strategy work. This synthesis aligns our approach with current hybrid-metaheuristic advances and extends existing techniques.

In short, the shortcomings of classical graph-based, potential-field, and sampling planners in cluttered or time-varying airspaces have moved the field's focus toward metaheuristic optimization—first with single paradigms such as GA, PSO, ACO, DE, and WOA, and now with carefully engineered hybrids that blend their complementary strengths. These bio-inspired algorithms furnish the flexibility and computational efficiency needed for modern UAV missions, and current research is pushing the envelope further by integrating adaptive control, learning components, and data-driven models. As UAV operations become more complex, the field is moving toward hybrid and ML-augmented frameworks that can meet real-time, high-dimensional planning demands.

3. Problem Statement

UAV path planning in dynamic environments involves determining an optimal, collision-free trajectory from a starting point to a goal, adapting to real-time environmental changes. Formally, let $p(t)=\left[x(t),y(t),z(t)\right]$ denote the UAV's position at time t. The UAV must navigate from an initial position $p(0)=p_{\text{start}}$ to a target position $p(T)=p_{\text{goal}}$

within a 3D space containing obstacles.

A feasible path $p(t)$ remains collision-free for all $t \in [0,T]$ and adheres to the vehicle's kinematic limits and environmental constraints. This path-planning problem is known to be NP-hard, with complexity growing rapidly as the environment and decision space expand. In dynamic settings, obstacles may move unpredictably, so the UAV must replan in real time to remain safe. The goal is to determine a trajectory $p(t)$ that optimally balances multiple criteria while guaranteeing no collisions at any time despite the changing environment.

Beyond optimizing distance or threat avoidance alone, real-world UAV navigation requires balancing several competing objectives. The planner must minimize length, maximize smoothness and safety, and respect energy, time, and altitude constraints. Key objectives include:

- Distance or path length (*L*): total flight distance traveled. Shorter paths generally reduce flight time and energy consumption.
- Smoothness (*S*): path curvature or turning effort. Large heading changes are undesirable; smoother trajectories (lower curvature) improve flight stability and fuel efficiency
- Threat/Risk (*R*): exposure to danger zones or obstacles. This can be quantified by the path's intersection with threat regions – e.g. overlapping a forbidden zone incurs a high cost. A safe path keeps a large clearance from obstacles to reduce collision risk.
- Energy/Fuel Consumption (*E*): the total energy or fuel expended. This is influenced by distance, speed, and maneuvers – e.g. aggressive turns or rapid climbs burn more energy. Prior studies include energy as a key component of the fitness function alongside distance and risk. For example, a path with excessive length or frequent altitude changes will increase power usage.
- Time of Flight (*T*): total mission time. Faster arrival is often critical (e.g. in surveillance or delivery missions). Time is related to distance and speed; minimizing T may conflict with minimizing E (since higher speeds reduce time but consume more energy).
- Navigational Altitude (*H*): the altitude profile along the route. Optimal altitude can improve efficiency and safety – for instance, flying higher might avoid ground obstacles but could incur stronger winds or airspace restrictions. Thus, the planner might aim to maintain an altitude that balances terrain clearance, energy efficiency, and threat exposure (e.g. staying below radar detection altitude).

These objectives form a multi-objective optimization problem. We can denote the objective vector as:

$$F(p(t)) = [L(p), S(p), R(p), E(p), T(p), H(p) \tag{1}$$

where each component quantifies one aspect of path quality. The aim is to simultaneously minimize all components (or achieve an optimal trade-off among them). In practice, one may combine these via weighted sum or prioritize them using Pareto optimality. For instance, a single aggregated cost J can be defined as a weighted sum of the major costs:

$$J(p) = w_1 L(p) + w_2 S(p) + w_3 R(p) + w_4 E(p) + w_5 T(p) + w_6 H(p) \tag{2}$$

The weights ($w_i > 0$) allow for mission-specific tuning, balancing the competing goals. The smoothness cost $S(p)$ serves as a direct proxy for kinodynamic feasibility; by penalizing high curvature, the algorithm inherently favors paths with gentle turns that respect a UAV's physical limits on acceleration and angular velocity. The threat cost $R(p)$ is a versatile component that enforces collision avoidance with physical objects and can be used to define and enforce adherence to virtual no-fly zones by modeling them as obstacles with an exceptionally high cost. The energy cost $E(p)$ directly addresses battery limitations by penalizing not just distance but also acceleration-heavy maneuvers, which are known to drain power more rapidly. Minimizing cost function $J(p)$ therefore yields a path that is efficient, safe, and physically achievable.

The objective of UAV path planning is to identify a path, denoted Path_i, that minimizes a cost function. The path is typically represented as a sequence of waypoints $M_{i,j}$ (where $j = 1, 2, \ldots, n$) in 3D space, with coordinates $(x_{i,j}, y_{i,j}, z_{i,j})$, where *i* indexes the *i*th path and *j* the *j*th waypoint along that path.

The path length cost, denoted as $L(p)$, represents the total distance traveled. In a continuous formulation it is given by:

$$L(p) = \int_0^T |\dot{p}(t)| dt \tag{3}$$

where $\dot{p}(t)=\frac{dp(t)}{dt}=\left[\frac{dx(t)}{dt},\frac{dy(t)}{dt},\frac{dz(t)}{dt}\right]$ represents the instantaneous velocity of the UAV. This integral captures the accumulation of infinitesimal distances traveled over time, providing a precise measure of the total flight path length. This formula quantifies the total distance a UAV travels along its continuous trajectory. In practical implementations, the continuous trajectory is often approximated by a sequence of discrete waypoints. The discrete approximation of the path length cost is then computed as:

$$L(\text{Path}_i)=\sum_{j=1}^{n-1}|M_{i,j+1}-M_{i,j}| \tag{4}$$

with the Euclidean distance between consecutive waypoints defined as:

$$\left|M_{i,j+1}-M_{i,j}\right|=\sqrt{\left(x_{i,j+1}-x_{i,j}\right)^2+\left(y_{i,j+1}-y_{i,j}\right)^2+\left(z_{i,j+1}-z_{i,j}\right)^2} \tag{5}$$

This discrete formula approximates the continuous path length by summing the straight-line distances between each pair of successive waypoints, which is practical for numerical optimization and simulation purposes. Discrete waypoint approximation is used due to its computational simplicity, which enables rapid real-time UAV path recalculations necessary for dynamic environments. Such an approach is particularly suited for realistic implementations where computation speed is crucial.

Smoothness, $S(p)$, quantifies how gently the path changes direction and altitude. A smoother path is generally preferred because it minimizes sharp turns or abrupt altitude changes, which are often energy-inefficient and can be challenging for the UAV's control system. In a continuous formulation, a common measure of smoothness is the integral of the squared curvature of the trajectory:

$$S(p)=\int_0^T \kappa(t)^2\,dt \tag{6}$$

where the curvature $\kappa(t)$ is defined by:

$$\kappa(t)=\frac{\left|\dot{p}(t)\times\ddot{p}(t)\right|}{|\dot{p}(t)|^3} \tag{7}$$

This formulation captures the rate of change of the direction of the velocity vector. Squaring the curvature ensures that larger deviations (i.e., sharper turns) contribute disproportionately to the cost, thereby encouraging smoother trajectories. For a discretized path the smoothness cost can be computed by considering the turning angles between consecutive path segments. Given three consecutive waypoints, one may compute the steering angle:

$$\psi_{i,j}=\arccos\left(\frac{\overrightarrow{v_{i,j}}\cdot\overrightarrow{v_{i,j+1}}}{\left|\overrightarrow{v_{i,j}}\right|\left|\overrightarrow{v_{i,j+1}}\right|}\right) \tag{8}$$

where $\overrightarrow{v_{i,j}}=M_{i,j+1}-M_{i,j}$ and $\overrightarrow{v_{i,j+1}}=M_{i,j+2}-M_{i,j+1}$. In addition, the change in the vertical direction (or the climb angle) is quantified by computing the climb angle $\theta_{i,j}$ at a waypoint, using the difference in the altitude component relative to the horizontal distance:

$$\theta_{i,j}=\arctan\left(\frac{z_{i,j+1}-z_{i,j}}{\sqrt{\left(x_{i,j+1}-x_{i,j}\right)^2+\left(y_{i,j+1}-y_{i,j}\right)^2}}\right) \tag{9}$$

The discrete smoothness cost is typically formulated as a weighted sum of the total turning angles and the differences in consecutive climb angles:

$$S(\text{Path}_i) = \sigma_1 \sum_{j=1}^{n-2} \psi_{i,j} + \sigma_2 \sum_{j=2}^{n-1} \left|\theta_{i,j} - \theta_{i,j-1}\right| \tag{10}$$

where σ_1 and σ_2 are weighting coefficients that determine the relative importance of the turning (steering) and vertical smoothness components, respectively. This formulation ensures that both horizontal and vertical smoothness are penalized if the changes are abrupt, thus promoting a path that is gentle and feasible for UAV flight.

Threat cost, denoted by $R(p)$, penalizes the proximity of the UAV to obstacles and computed by evaluating each segment's minimum distance to obstacle α:

$$R(\text{Path}_i) = \sum_{j=1}^{n-1} \sum_{\alpha=1}^{M} K_\alpha \left(M_{i,j}, M_{i,j+1}\right) \tag{11}$$

with K_α is defined as:

$$K_\alpha\left(M_{i,j}, M_{i,j+1}\right) = \begin{cases} 0, \ if \ d_\alpha\left(M_{i,j}, M_{i,j+1}\right) > S + D + R_\alpha \\ \left(S + D + R_\alpha\right) - d_\alpha\left(M_{i,j}, M_{i,j+1}\right), if \ D + R_\alpha < d_\alpha\left(M_{i,j}, M_{i,j+1}\right) \le S + D + R_\alpha \\ \infty, if \ d_\alpha\left(M_{i,j}, M_{i,j+1}\right) \le D + R_\alpha \end{cases} \tag{12}$$

where $d_\alpha\left(M_{i,j}, M_{i,j+1}\right)$ is the minimum distance between the path segment and obstacle α, as illustrated in Figure 1.

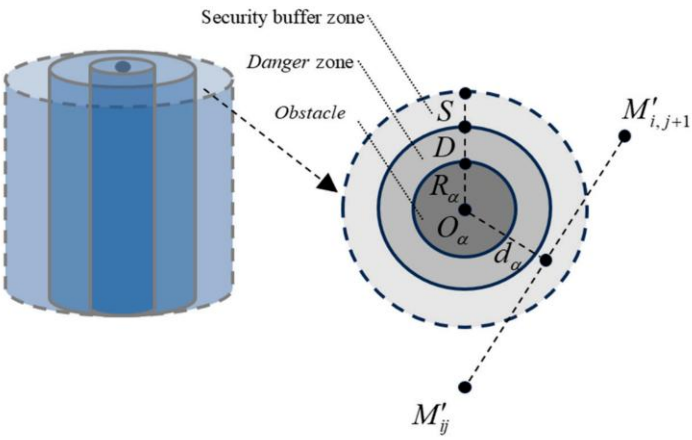

Fig.1. Obstacle thread model

Altitude cost, $H(p)$, ensures the UAV remains within a desirable altitude range (shown on Fig. 2). In a continuous formulation, if $z_d(t)$ represents the desired altitude profile, then

$$H(p) = \int_0^T \left|z(t) - z_d(t)\right| dt \tag{13}$$

For discrete paths, assuming an optimal altitude $z_{\text{avg}} = \frac{z_{min} + z_{max}}{2}$, the cost at each waypoint is defined as:

$$H_{i,j} = \begin{cases} \left|z_{i,j} - z_{\text{avg}}\right|, \text{if } z_{min} \le z_{i,j} \le z_{max} \\ \infty, \text{otherwise} \end{cases} \tag{14}$$

so that:

$$H(\text{Path}_i) = \sum_{j=1}^{n} H_{i,j} \tag{15}$$

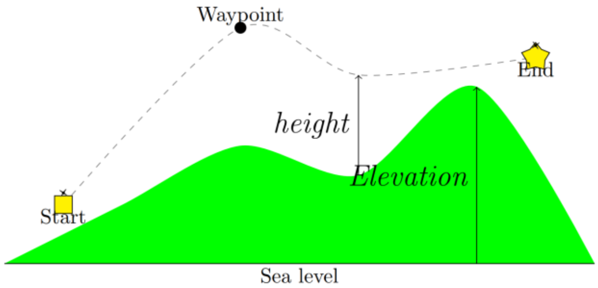

Fig.2. UAV altitude cost

Energy consumption cost, $E(p)$, reflects both the distance and the dynamic effort required to traverse the path. Continuously, it may be modeled as:

$$E(p)=\int_0^T \mathrm{E}\left(p(t),\dot{p}(t),\ddot{p}(t)\right)dt \tag{16}$$

where $\mathrm{E}(\cdot)$ is an energy model derived from the UAV's dynamics. In the discrete case, a simplified energy cost combines the travel distance and the intensity of acceleration:

$$E(\text{Path}_i)=\sum_{j=1}^{n-1}\left[\lambda_1\left|M_{i,j+1}-M_{i,j}\right|+\lambda_2\left|M_{i,j+1}-2M_{i,j}+M_{i,j-1}\right|\right] \tag{17}$$

with λ_1 and λ_2 being constants that weight the contributions of travel distance and acceleration, respectively, and the term $\left|M_{i,j+1}-2M_{i,j}+M_{i,j-1}\right|$ approximates the second derivative of the position, representing the acceleration between waypoints.

Finally, the flight time cost, $T(p)$, is directly related to the path length when assuming a constant UAV speed v. In the continuous model, flight time is simply T, whereas discretely it can be approximated as:

$$T(\text{Path}_i)=\frac{1}{v}\sum_{j=1}^{n-1}|M_{i,j+1}-M_{i,j}| \tag{18}$$

Combining these components, the overall multi-objective cost function for a candidate path Path_i is expressed as:

$$J(\text{Path}_i)=w_1 L(\text{Path}_i)+w_2 S(\text{Path}_i)+w_3 R(\text{Path}_i)+w_4 H(\text{Path}_i)+w_5 E(\text{Path}_i)+w_6 T(\text{Path}_i) \tag{19}$$

where each term is defined in both its continuous and discrete forms as described above and is weighted by a factor $(w_1,\ldots,w_6)$ reflecting its importance in the mission. This multi-objective cost formulation allows flexible trade-offs; for example, one can emphasize safety by increasing w_3 (risk) and w_2 (smoothness) or prioritize efficiency by emphasizing w_1, w_5, w_6. Notably, our cost function is more comprehensive than those in some prior works, covering a wide array of factors, and can be extended by adding new terms or adjusting weights to meet different mission requirements. In our simulations, the weight values are chosen to ensure all constraints (distance, obstacle clearance, altitude, etc.) are respected and to mimic realistic flight priorities. For instance, relatively higher weights on risk and smoothness (w_2,w_3) encourage safe paths that detour around no-fly zones and avoid sudden turns, while moderate weights on length and time keep the route efficient. A candidate path is considered feasible only if it does not violate hard constraints (colliding with an obstacle or entering a forbidden zone); infeasible paths are assigned a very high cost via the risk term or are disqualified. The cost function guides the optimizer toward low-cost, and thus safe and efficient, trajectories.

To make the multi-objective cost J comparable across heterogeneous terms, each raw component is first normalised to the interval [0,1]. The weight vector $w=(0.25,0.25,0.2,0.1,0.1,0.1)$ is then applied to the normalised terms (length, smoothness, risk, altitude, energy, time), reflecting a mission profile that prioritises short, smooth and safe trajectories while still accounting for endurance and punctual arrival. The coefficients were calibrated in a preliminary sweep: length, smoothness and risk were incrementally varied in 0.05 steps until the resulting paths achieved both $\geqslant$15m obstacle clearance and $\leqslant$ 5° mean curvature, after which the remaining weights were proportionally scaled.

4. Methodology

UAV path planning in complex 3D environments with obstacles is a challenging global optimization problem. Purely exploiting or purely exploring algorithms often struggle to find near-optimal paths within reasonable time and computational limits. To address this, we propose a hybrid two-phase approach that leverages the complementary strengths of DE and an E-WOA. Hybrid metaheuristics are widely used to balance exploration and exploitation, typically outperforming single-algorithm approaches. In Phase 1, DE is utilized for global exploration of the search space, leveraging its robust global search capability. In Phase 2, E-WOA is applied to intensify the search around promising regions, utilizing the WOA's exploitation proficiency to refine solutions. DE is robust, has few parameters, and handles multi-modal landscapes well. However, DE's local search (exploitation) efficiency is limited if used alone. The WOA is recognized for its straightforward implementation and effective exploitation, inspired by humpback whale foraging behavior. However, the standard WOA may experience premature convergence due to limited exploration in later iterations. Therefore, we integrate DE and an improved WOA in a synergistic manner: the DE phase prevents the search from getting trapped in poor regions by global exploration, while the subsequent E-WOA phase accelerates convergence on a high-quality path by local exploitation. The WOA component is enhanced with several mechanisms to further remedy WOA's inherent shortcomings. These enhancements – including real-time boundary handling, quasi-oppositional learning, and adaptive search dynamics – bolster the algorithm's global optimization ability and convergence speed [1]. In summary, DE + E-WOA first diversifies the population and then intensifies the search around the best candidates, striking an effective balance between exploration and exploitation crucial for complex UAV path planning. The transition from DE to E-WOA occurs upon detection of stagnation or reaching a predefined exploration threshold. Specifically, the best candidate solutions obtained from the final DE iteration serve as initial solutions for the subsequent E-WOA exploitation phase, allowing intensive refinement in the promising regions identified previously.

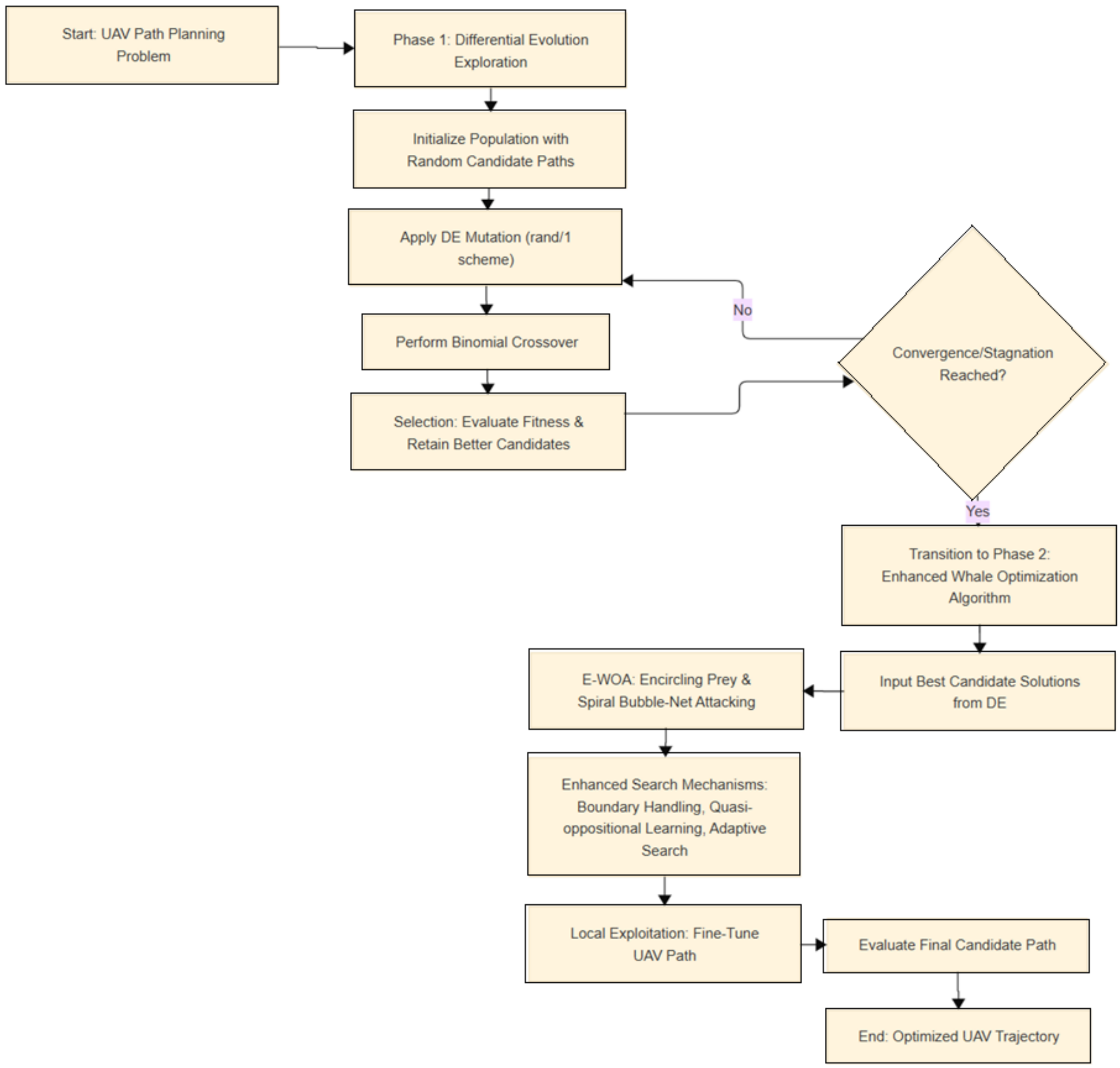

Fig.3. Flowchart representing the two-phase hybrid optimization approach

To clearly illustrate the workflow and enhance understanding of the proposed hybrid DE + E-WOA optimization framework, a structured flowchart is presented in Figure 3. The flowchart summarizes the two-phase approach: global exploration with DE followed by local exploitation with E-WOA. By explicitly delineating key steps—including population initialization, mutation and crossover operations, adaptive transitioning between phases, and enhanced search mechanisms—this visual representation underscores the rigorous methodological integration essential for achieving optimized UAV trajectories in dynamic and obstacle-rich three-dimensional environments.

4.1. Phase 1: Exploration with Differential Evolution

DE is a population-based evolutionary algorithm well-known for its extensive global exploration capability, generating new candidate solutions through scaled differences between existing solutions [54]. In our approach, each individual (candidate UAV path) is encoded as a vector of decision variables (e.g., waypoint coordinates or control points). The population of size N (denoted $P = x_1, x_2, \ldots, x_n$) is initially sampled randomly within the feasible search space (respecting flight constraints). Let $x_i^{(g)} = \left[x_{i,1}^{(g)}, x_{i,2}^{(g)}, \ldots, x_{i,D}^{(g)} \right]$ represent the i th solution at generation g, where D is the problem dimension. DE iteratively improves the population through mutation, crossover, and selection operators, as follows:

- Mutation (Diversification). For each target individual $x_i^{(g)}$, DE generates a donor vector $v_i^{(g)}$ by combining other individuals. A common strategy used in this phase is the "rand/1" scheme, chosen for its exploration strength. In DE/rand/1, three distinct random indices $r_1, r_2, r_3 \neq i$ are selected from $1, \ldots, N$. The donor is produced by:

$$v_i^{(g)} = x_{r_1}^{(g)} + F \cdot \left(x_{r_2}^{(g)} x_{r_3}^{(g)} \right) \tag{20}$$

where $F \in (0,1)$ is a scaling factor that controls the magnitude of the perturbation. This promotes exploration by extrapolating from the difference of two random individuals, probing new areas of the search space. (In our implementation, we set *F* to a moderate value like 0.5-0.8 as recommended in literature to balance diversity and stability.) Other mutation strategies (e.g., current-to-best or best/2 variants) could be employed, but we prioritize those injecting randomness to maximize coverage of the search space in Phase 1.

- Crossover (Recombination). The donor vector $v_i^{(g)}$ is then mixed with the target $x_i^{(g)}$ to create a trial vector $u_i^{(g)}$. We use binomial crossover: for each component $j \in 1, \ldots, D$,

$$u_{i,j}^{(g)} = \begin{cases} v_{i,j}^{(g)}, \text{if } r_j \leq CR \text{ or } j = j_{\text{rand}} \\ x_i^{(g)}, \text{otherwise} \end{cases} \tag{21}$$

where r_j is a uniform random number in [0,1] for the j th dimension, $CR \in [0,1]$ is the crossover probability (typically high, e.g. *CR*=0.9), and j_{rand} is a randomly chosen index to ensure at least one gene comes from the donor. Equation means each trial solution inherits most components from the donor v_i (promoting exploration) but retains some from the current solution x_i (preserving inherited good traits), thereby maintaining a balance between exploration and exploitation at the individual level.

- Selection (Survival of Fittest). After crossover, the trial individual $u_i^{(g)}$ is evaluated. DE uses a greedy selection: if the trial's fitness is better (lower cost) than that of the target, it replaces the target in the next generation; otherwise, the original target is retained:

$$x_i^{(g+1)} = \begin{cases} u_i^{(g)}, \text{if } J\left(u_i^{(g)}\right) \leq J\left(x_i^{(g)}\right) \\ x_i^{(g)}, \text{otherwise} \end{cases} \tag{22}$$

This elitist selection ensures the population never degrades in quality – the best-so-far solution is monotonically improving or at least maintained across generations.

4.2. Phase 2: Exploitation with Enhanced Whale Optimization Algorithm

In this phase, a refined WOA is employed to exploit the promising regions of the search space identified in Phase 1.

The standard WOA is a nature-inspired optimizer mimicking the bubble-net hunting behavior of humpback whales. It is well-suited for exploitation due to its fast convergence and simple control parameters. However, the basic WOA can suffer from premature convergence or slow fine-tuning in complex terrains. To overcome these issues, we adopt an Enhanced WOA that integrates several improvements into the standard algorithm, these mechanisms are designed to preserve solution feasibility, intelligently expand the search space, and adaptively escape from local optima.

Real-time boundary handling is invoked immediately after any position update. If a coordinate x_j of a candidate whale falls outside its permissible bounds $\left[L_j, U_j\right]$, it is clamped according to $x_j = \max\left(L_j, \min\left(x_j, U_j\right)\right)$, ensuring that every agent always represents a physically admissible UAV path. This lightweight repair step eliminates wasted fitness evaluations on infeasible solutions and, by preserving population size, maintains steady selection pressure throughout the run.

To forestall diversity loss, quasi-oppositional learning is applied periodically (every ten) to a randomly selected subset of whales. For a current position X in a space bounded by L and U, we first compute its opposite point $X_{\text{op}} = L + U - X$. A quasi-opposite candidate is then drawn along the line segment connecting the search-space center $M = \frac{L+U}{2}$ and X_{op}:

$$X_{\text{qo}} = M + r\left(X_{\text{op}} - M\right) \tag{23}$$

Here, r is drawn uniformly at random from the interval [0,1]. If X_{qo} yields a lower cost than X, it immediately replaces X in the population. This targeted, geometry-aware mutation encourages exploration of distant yet promising regions without disrupting the algorithm's overall convergence trajectory.

Finally, an adaptive stagnation-response mechanism tracks the number of consecutive generations in which the global best cost fails to improve. Once this counter exceeds a threshold $S_{max} = 15$, the algorithm injects controlled perturbations. With equal probability, it either partially resets the convergence factor a to 1.5, temporarily forcing $|A| > 1$ for many whales and thereby re-opening a broader exploratory radius. The counter is then zeroed, and normal exploitation resumes. This feedback loop allows E-WOA to escape shallow local minima without permanently sacrificing the intensified search around previously identified high-quality regions.

In the following, we detail the mathematical formulation of E-WOA, the algorithmic enhancements, the implementation, and the chosen parameters that balance exploration and exploitation in this phase.

- Encircling prey (shrinking encirclement). In WOA, whales assume the current best solution is the "prey" and update their positions by encircling this best whale. For a given whale at position $X(t)$ (at iteration t) and the current best position $X^*(t)$, the encircling behavior is modeled as:

$$D = \left|C \cdot X^*(t) - X(t)\right| \tag{24}$$

$$X(t+1) = X^*(t) - A \cdot D \tag{25}$$

where A and C are coefficient vectors and D is the distance (absolute difference) between the whale and the prey. The coefficients A and C are defined as:

$$A = 2a \cdot r_1 - a \tag{26}$$

$$C = 2 \cdot r_2 \tag{27}$$

with r_1, r_2 being random vectors in $[0,1]^n$ (for an n-dimensional search space). The parameter a linearly decreases from 2 to 0 over the course of iterations (to transition from exploration to exploitation). For example, a common schedule is:

$$a(t) = 2 - 2\frac{t}{T_{max}} \tag{28}$$

where T_{max} the maximum number of iterations. As a decreases, the magnitude of A diminishes, which means whales gradually reduce their search radius and focus on local exploitation around the best solution.

- Spiral bubble-net attacking. Humpback whales also perform a spiral motion to herd prey. In WOA, each whale can move along a logarithmic spiral towards the best whale with a certain probability. The spiral update is given by:

$$D' = \left\| X^*(t) - X(t) \right\| \tag{29}$$

$$X(t+1) = D' \cdot e^{b \cdot l} \cdot \left(\cos(2\pi l)\right) + X^*(t) \tag{30}$$

Here D' is the Euclidean distance between the i th whale and the best whale, b is a constant defining the spiral's tightness (commonly $b=1$), and l is a random number in [-1,1]. This equation generates a spiral-shaped path from $X(t)$ towards $X^*(t)$. By adding this to $X^*(t)$, the whale's new position $X(t+1)$ lies on the spiral converging around the best solution. The spiral search introduces a different exploitation pattern that can fine-tune the position relative to the current best.

- Exploitation–exploration switching. The WOA algorithm dynamically switches between encircling behavior (exploitation) and random exploration, based on the value of parameter $|A|$. If $|A|<1$, whales are drawn close to the prey and exploit the best region; if $|A| \geq 1$, a whale searches globally. Furthermore, WOA uses a random probability p to choose between the shrinking encircling strategy and the spiral strategy when exploiting. Specifically, for each whale, if $p<0.5$, the position is updated by encircling, and if $p \geq 0.5$, the spiral equation is used. This 50–50 selection (along with random l in the spiral) adds stochasticity to exploitation, preventing a deterministic trap around the best position. On the other hand, when $|A| \geq 1$, the whale performs a random search by abandoning the current best and choosing a random reference whale X_{rand} from the population. In that case, the position update uses a similar equation:

$$D = \left| C \cdot X_{\text{rand}}(t) - X(t) \right| \tag{31}$$

$$X(t+1) = X_{\text{rand}}(t) - A \cdot D \tag{32}$$

which drives the whale towards a random individual instead of the global best. This mechanism maintains exploration ability in the algorithm's early stages or whenever the whales are far from the prey.

- Enhanced search mechanisms in E-WOA. The E-WOA retains the above WOA operations but augments them with additional steps to improve performance. During the exploitation phase, we bias the algorithm towards intensification. This can be done by adjusting the parameter schedule or probability p to spend more effort on the local search around the best solution. For instance, we set the initial a to a smaller value than 2 (since Phase 1 already identified a good region), ensuring that $|A|<1$ more frequently. This keeps the whales in exploitation mode for a larger portion of iterations, homing in on the optimum. At the same time, E-WOA introduces new update rules to escape local optima when needed. One such enhanced search step is to perturb the best solution slightly and share that information with the swarm. For example, with a certain small probability, the best whale's position $X^*(t)$ can be adjusted as:

$$X^*(t) \leftarrow X^*(t) + \mu \cdot \left(U - X^*(t)\right) \tag{33}$$

where μ is a random scalar in [0,1] and U is a uniformly sampled random position in the neighborhood or entire space. This creates a mutated best candidate, which is then used for encircling/spiring if it offers improvement. Such an enhanced search mechanism diversifies the exploitation process, giving E-WOA the ability to intensely exploit near the current best while still having a fallback strategy to find a better peak if the current region is suboptimal.

One of the primary enhancements is the implementation of real-time boundary handling. In UAV path planning, candidate solutions must strictly comply with spatial and operational constraints; hence, each updated position is immediately verified against the predefined limits. Any coordinate that exceeds these bounds is promptly corrected—typically through clamping or reflective adjustment—to ensure that the solution remains feasible. This mechanism not only preserves the validity of candidate paths throughout the iterative process but also avoids unnecessary evaluations of infeasible solutions, thereby enhancing the algorithm's overall efficiency and reliability.

Another significant improvement in E-WOA is the incorporation of quasi-oppositional learning. This strategy is predicated on the observation that for any given candidate solution, its opposite (computed by reflecting each coordinate with respect to the search space boundaries) might be closer to the global optimum. However, rather than considering the exact opposite, E-WOA generates a quasi-opposite candidate by blending the original solution with its opposite using a random interpolation factor. This controlled diversification allows the algorithm to probe mirrored regions of the search space without drastically diverging from promising areas. The quasi-oppositional approach has been shown to accelerate convergence by effectively expanding the search horizon and helping the swarm to escape local minima.

Additionally, E-WOA employs adaptive search mechanisms that dynamically balance the exploration and exploitation processes during the optimization run. In contrast to the standard WOA, where parameter adjustment (such as the linearly decreasing convergence factor) is predetermined, E-WOA continuously monitors the progress of the search. When the algorithm detects stagnation—characterized by negligible improvement in the best solution over a number of iterations—it responds by adjusting its control parameters. For example, the convergence factor may be temporarily increased to reintroduce exploratory moves, or a mutation-like perturbation can be applied to diversify the candidate solutions. This real-time adaptation ensures that the algorithm remains flexible and responsive to the evolving landscape of the optimization problem, significantly reducing the likelihood of premature convergence. Together, these three mechanisms preserve feasibility, sustain diversity, and adapt search pressure, yielding refined UAV paths in cluttered 3D spaces.

5. Simulations and Results

This section presents a comprehensive simulation study evaluating the performance of our proposed hybrid algorithm, DE + E-WOA, for UAV path planning in three-dimensional environments containing static obstacles. The hybrid approach integrates DE for initial global exploration, followed by an E-WOA for intensive local exploitation. The resulting algorithm is benchmarked against standard WOA, PSO, DE, Symbiotic Organism Search (SOS), and GWO using the overall cost function J defined in Equation 19.

All algorithms minimize the same cost function J under identical settings, enabling fair comparison of path quality and feasibility. By including both classical algorithms (PSO, DE) and a relatively recent one along with WOA, we cover a diverse set of approaches: particle dynamics, evolutionary mutation-based, and ecological strategy.

We simulate UAV path planning in a 3D environment of size 1000 × 1000 × 500 (in arbitrary distance units, e.g., meters). The environment was populated with a constrained set of four distinct static obstacles, comprising 2-4 cylinders and rectangular prisms—randomly distributed to create narrow corridors and blind spots. These obstacles were strategically placed to block the direct line-of-sight trajectory between the start and end points. This configuration requires the optimization algorithms to identify a circuitous, non-linear path. This setup robustly tests each algorithm's ability to balance length minimization and obstacle avoidance.

We represent a path as a series of waypoints (control points) in 3D, connected by straight line segments. A smoothing procedure (such as fitting a B-spline curve through these waypoints) is applied to ensure the path is flyable (continuous curvature). If a smoothed path segment violates obstacle clearance, the cost function heavily penalizes it (via the risk term) or the path is discarded. We set a buffer zone around each obstacle to ensure a safe clearance distance, thereby increasing the risk cost sharply if a path comes too near an obstacle. All algorithms use a population of 50 candidate solutions and run for a maximum of 200 generations (iterations). All algorithms start with the same initial population of random paths. Each candidate path's feasibility and cost are evaluated using the cost function. A generation consists of updating all candidates by the algorithm's rules and then evaluating new costs. For statistical reliability, results are averaged over 30 independent runs on an Intel Core i7-14700KF with 16 GB RAM using Python.

The comparative evaluation was based on a set of specific quantitative metrics designed to measure different aspects of algorithm performance. The primary metrics include the Average Final Cost, which is the mean value of the objective function J at the final iteration, indicating overall path quality. The Standard Deviation of this cost was also measured to assess the consistency and reliability of each algorithm. The Average Path Length provides a direct measure of flight efficiency, corresponding to time and energy savings. To address practical applicability, the Average Computation Time was recorded, measuring the total time to complete a single run of 200 iterations. A summary of the aggregated results over 30 independent runs is presented in Table 1.

Table 1. Comparative performance of algorithms

Algorithm	Avg. Final Cost (J)	Std. Dev. of Cost	Avg. Path Length (m)	Avg. Comp. Time (s)
DE + E-WOA	293.7	14.2	1351.2	2.35
PSO	418.9	44.5	1658.3	1.87
DE	406.1	38.9	1614.5	2.47
SOS	431.3	58.2	1705.4	2.84
WOA	362.4	26.1	1555.8	3.41
GWO	343.5	21.8	1521.7	2.23

The results in Table 1 clearly illustrate the enhanced performance of the proposed DE + E-WOA framework. Our hybrid method achieved an average final cost of 293.7, which is 14.5% lower than that of the next-best algorithm, GWO, and substantially lower than other tested algorithms. This indicates a more optimal balance between path length, smoothness, and obstacle clearance. The standard deviation of the final cost is also the lowest. In terms of efficiency, the DE + E-WOA generated paths with an average length of 1351.2 m, which is 11.2% shorter than GWO's paths and over 15% shorter than those produced by DE, WOA, and PSO. A crucial aspect of UAV path planning is the computational overhead, which dictates the method's viability for real-world applications. As detailed in Table 1, the DE + E-WOA framework required an average of 2.35 seconds to complete a full optimization run. Although PSO and GWO exhibited a slightly faster average time, this marginal speed advantage was accompanied by a severe deficit in solution quality and reliability. In contrast, our hybrid approach is substantially faster than the standard WOA, while delivering markedly enhanced path solutions. An average computation time on a non-specialized computing platform confirms that the DE + E-WOA is well-suited for near-real-time applications This performance level is adequate for missions where strategic replanning is required in response to newly detected threats or changing mission objectives (e.g., replanning every 5-10 seconds). While this may not be sufficient for highly reactive, low-level control (which demands sub-second responses), it is fast enough for the vast majority of operational UAV scenarios, including long-range surveillance, precision agriculture, and infrastructure inspection. Furthermore, an optimized C++ implementation on dedicated UAV hardware would substantially reduce this computation time, further expanding the range of missions for which it is practical. These results indicate that the proposed framework is both effective and computationally practical.

Trajectory quality is paramount. In Figure 4 we compare the final, smoothed paths obtained by our hybrid DE + E-WOA algorithm with others. The filled cylinders in the 3D view (left panel) and their corresponding top-down footprints in the 2D view (right panel) represent the obstacles with their prescribed buffer zones. These obstacles enforce the constraint that all feasible paths must maintain a safe clearance.

The DE + E-WOA path, is notably the smoothest and most direct route. It elegantly curves around the obstacles, maintaining ample clearance from each one, which minimizes both the risk and the additional energy and time costs associated with steep maneuvers. In contrast, the paths produced by the other algorithms appear less smooth indicating that these methods either compromise on smoothness or sacrifice clearance to minimize path length. For instance, the PSO path shows longer path and closely approaches obstacle boundaries, while the DE and SOS paths, although feasible, display unnecessary detours and less uniform curvature. The GWO path, although relatively stable, does not match the low turning angles of the DE + E-WOA route. In the top-down view it is evident that the DE + E-WOA trajectory maintains a clear buffer around the obstacles, whereas the alternative paths tend to thread closer to the obstacle footprints, potentially increasing the risk of collision.

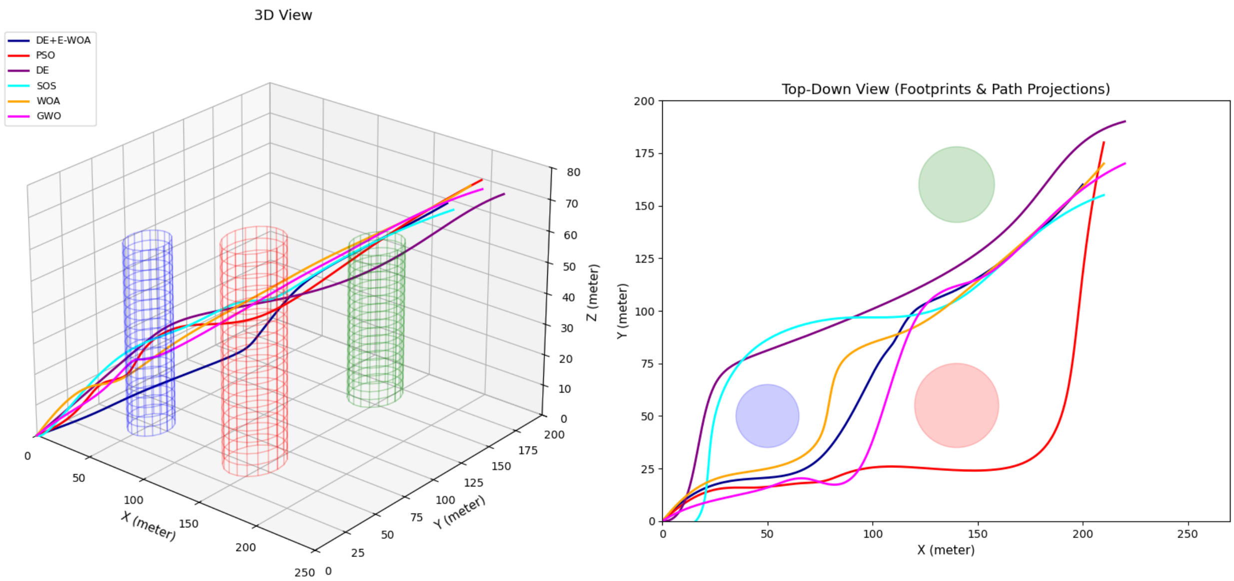

Fig.4. Path planning and obstacle avoidance map

The convergence characteristics of each algorithm specifically, the speed and effectiveness with which they approach an optimal solution are quantitatively assessed. Figure 5 illustrates a representative case the convergence curved of each method. We see that the DE + E-WOA drives the cost down steeply within the first ~10 iterations, reaching a near-optimal solution quickly, whereas standard algorithms converge slower or to higher plateaus. In contrast, the PSO and DE algorithms initially demonstrate rapid reductions in cost, but then level off at higher values, suggesting convergence to suboptimal solutions.

This trend was observed across our experiments: E-WOA typically converges to a stable solution in under half the allotted iterations, whereas standard WOA, PSO, and DE often require many more iterations and still end up with slightly higher cost values. SOS sometimes finds good initial solutions (due to its aggressive random search phase) but then can stagnate; in our tests SOS frequently got trapped in local minima, indicated by flat convergence curves at non-optimal costs. Faster convergence saves computation time and indicates that E-WOA exploits the search space more effectively. We attribute E-WOA's rapid convergence to its integrated strategies that intensify the search around promising solutions while still injecting diversity (preventing premature convergence). Overall, E-WOA demonstrated superior convergence behavior, reliably finding high-quality paths faster than the alternatives.

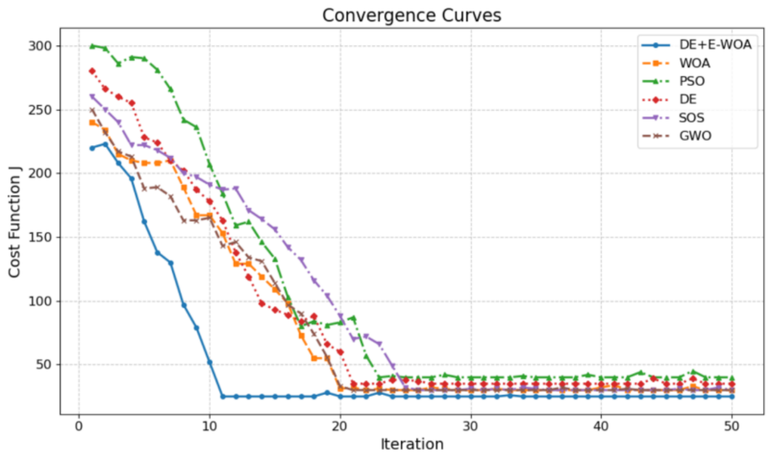

Fig.5. Algorithms convergence curves

6. Conclusions

This study introduces a novel hybrid optimization framework that integrates DE with an E-WOA to address the challenging multi-objective UAV path planning problem in complex three-dimensional environments. The two-phase approach utilizes DE's global search capabilities to explore diverse candidate trajectories, followed by E-WOA's refined local exploitation to optimize promising solutions. This hybrid framework effectively overcomes the limitations of traditional path-planning methods and standalone metaheuristic algorithms. Enhancements in the E-WOA algorithm, such as real-time boundary handling, quasi-oppositional learning, and adaptive search dynamics, effectively prevent premature convergence, facilitating rapid optimization of high-quality paths.

Simulation results demonstrate that the proposed DE + E-WOA framework consistently outperforms conventional metaheuristic approaches such as standard DE, PSO, SOS, and GWO. The hybrid algorithm generates UAV trajectories that not only achieve lower overall costs across multiple objectives (including path length, smoothness, risk, energy consumption, and flight time) but also maintain obstacle clearance and safe navigation margins. Notably, the convergence analysis reveals that the hybrid method reaches near-optimal solutions in significantly fewer iterations, highlighting its potential for real-time applications in dynamic and cluttered environments.

This integration allows the algorithm to adapt its strategy to convergence progress, maintaining a balance between exploration and exploitation. Such adaptability is particularly crucial in dynamic environments where obstacles may appear unpredictably, and real-time decision-making is essential. Moreover, the framework's ability to generate smooth trajectories with minimal curvature not only enhances flight stability but also reduces energy consumption, addressing critical operational constraints in UAV missions. Compared with the benchmarks, DE + E-WOA produced shorter, smoother, and safer trajectories, indicating its promise for autonomous UAV navigation.

Although the DE + E-WOA framework marks an advancement, its practical implementation faces challenges that outline key directions for future research. Like most metaheuristics, the hybrid framework's performance depends on meticulous parameter tuning and may still converge to local optima in highly complex cost landscapes. This issue may intensify in cluttered or dynamic environments with rapidly shifting obstacles. Consequently, practical deployment will necessitate automated or adaptive tuning strategies to eliminate manual trial-and-error. Furthermore, this study was conducted within a deterministic, static environment, using path smoothness as a proxy for kinodynamic feasibility. This simplification omits real-world factors such as wind, which can markedly affect trajectory tracking and energy use. The current framework is also designed exclusively for single-agent planning and does not natively address the challenges of multi-UAV coordination.

These limitations directly inform a clear agenda for future work. Our current study addresses only single-UAV planning under kinematic constraints and static obstacles; extending the approach to account for weather or wind disturbances, fluid-dynamic effects on energy cost, multi-agent collision avoidance, probabilistic occupancy maps, or fully coupled swarm coordination will introduce further computational trade-offs that must be carefully managed. To bridge the gap between planning and control, the framework should be validated with a full dynamic UAV model and integrated with a model predictive control (MPC) backend, which would ensure the generated trajectories are truly flyable. Future work will extend the framework to multi-UAV coordination, incorporating collision-avoidance constraints and deconfliction strategies. For highly specialized applications, one could even envision integrating simplified computational fluid dynamics (CFD) into the cost function to avoid regions of high turbulence.

This approach, while essential for controlled comparative analysis and algorithm development, does not account for many of real-world factors that can affect flight performance. These include sensor noise, actuator latencies, unmodeled aerodynamic effects, and communication delays. Therefore, a crucial and necessary direction for future research is to bridge this 'sim-to-real' gap. The next logical phase of validation involves progressing to a Hardware-in-the-Loop (HIL) simulation, where the algorithm runs on its target embedded hardware and interacts with a high-fidelity physics simulator. The ultimate validation, however, will be the deployment and field testing of the framework on a physical UAV platform. Empirical validation is essential to demonstrate robustness and reliability under operational conditions.

In summary, the DE + E-WOA framework provides a powerful and computationally practical foundation for UAV path planning. By acknowledging its current boundaries and actively pursuing these future research directions, its potential can be fully realized, moving from a high-performing optimization algorithm to a truly robust and intelligent navigation system for autonomous aerial vehicles.

References

[1] S. Yin, J. Yang, L. Ma, M. Fu, and K. Xu, "An Enhanced Whale Algorithm for Three-Dimensional Path Planning for Meteorological Detection of the Unmanned Aerial Vehicle in Complex Environments," IEEE Access, vol. 12. Institute of Electrical and Electronics Engineers (IEEE), pp. 60039–60057, 2024. doi:10.1109/access.2024.3394055.

[2] Y. Li, A. Gao, H. Li, and L. Li, "An improved whale optimization algorithm for UAV swarm trajectory planning," Advances in Continuous and Discrete Models, vol. 2024, no. 1. Springer Science and Business Media LLC, Sep. 26, 2024. doi:10.1186/s13662-024-03841-4.

[3] Q. Wu, W. Tan, R. Zhan, L. Jiang, L. Zhu, and H. Wu, "GLBWOA: A Global–Local Balanced Whale Optimization Algorithm for UAV Path Planning," Electronics, vol. 13, no. 23. MDPI AG, p. 4598, Nov. 21, 2024. doi:10.3390/electronics13234598.

[4] Y. Dai, J. Yu, C. Zhang, B. Zhan, and X. Zheng, "A novel whale optimization algorithm of path planning strategy for mobile robots," Applied Intelligence, vol. 53, no. 9. Springer Science and Business Media LLC, pp. 10843–10857, Aug. 26, 2022. doi:10.1007/s10489-022-04030-0.

[5] H. Liu, Q. Chen, N. Pan, Y. Sun, Y. An, and D. Pan, "UAV Stocktaking Task-Planning for Industrial Warehouses Based on the Improved Hybrid Differential Evolution Algorithm," IEEE Transactions on Industrial Informatics, vol. 18, no. 1. Institute of Electrical and Electronics Engineers (IEEE), pp. 582–591, Jan. 2022. doi:10.1109/tii.2021.3054172.

[6] G. Hu, F. Huang, A. Seyyedabbasi, and G. Wei, "Enhanced multi-strategy bottlenose dolphin optimizer for UAVs path planning," Applied Mathematical Modelling, vol. 130. Elsevier BV, pp. 243–271, Jun. 2024. doi:10.1016/j.apm.2024.03.001.

[7] Y. Zhang, Y. Zhu, H. Li, and J. Wang, "A hybrid optimization algorithm for multi-agent dynamic planning with guaranteed convergence in probability," Neurocomputing, vol. 592. Elsevier BV, p. 127764, Aug. 2024. doi:10.1016/j.neucom.2024.127764.

[8] X. Zhang, L. Yang, Y. Li, W. Yang, and Y. Shi, "A nonlinear weighted whale optimization algorithm improved by cosine function for UAV path planning," IET Conference Proceedings, vol. 2024, no. 13. Institution of Engineering and Technology (IET), pp. 1001–1007, Jan. 2025. doi:10.1049/icp.2024.3022.

[9] M. Nikolaiev and M. Novotarskyi, "Comparative Review of Drone Simulators," Information, Computing and Intelligent systems, no. 4. Igor Sikorsky Kyiv Polytechnic Institute, pp. 79–98, Oct. 02, 2024. doi:10.20535/2786-8729.4.2024.300614.

[10] Z. Zhang, J. Zhu, and F. Nie, "A novel hybrid adaptive differential evolution for global optimization," Scientific Reports, vol. 14, no. 1. Springer Science and Business Media LLC, Aug. 24, 2024. doi:10.1038/s41598-024-70731-w.

[11] M. Hooshyar and Y.-M. Huang, "Meta-heuristic Algorithms in UAV Path Planning Optimization: A Systematic Review (2018–2022)," Drones, vol. 7, no. 12. MDPI AG, p. 687, Nov. 25, 2023. doi:10.3390/drones7120687.

[12] G. Sun, Y. Shang, K. Yuan, and H. Gao, "An Improved Whale Optimization Algorithm Based on Nonlinear Parameters and Feedback Mechanism," International Journal of Computational Intelligence Systems, vol. 15, no. 1. Springer Science and Business Media LLC, Jun. 10, 2022. doi:10.1007/s44196-022-00092-7.

[13] S. Fu, K. Li, H. Huang, C. Ma, Q. Fan, and Y. Zhu, "Red-billed blue magpie optimizer: a novel metaheuristic algorithm for 2D/3D UAV path planning and engineering design problems," Artificial Intelligence Review, vol. 57, no. 6. Springer Science and Business Media LLC, May 03, 2024. doi:10.1007/s10462-024-10716-3.

[14] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," Computer Communications, vol. 149. Elsevier BV, pp. 270–299, Jan. 2020. doi:10.1016/j.comcom.2019.10.014.

[15] Y. Zhao, Z. Zheng, and Y. Liu, "Survey on computational-intelligence-based UAV path planning," Knowledge-Based Systems, vol. 158. Elsevier BV, pp. 54–64, Oct. 2018. doi:10.1016/j.knosys.2018.05.033.

[16] J. Tisdale, Z. Kim, and J. Hedrick, "Autonomous UAV path planning and estimation," IEEE Robotics & Automation Magazine, vol. 16, no. 2. Institute of Electrical and Electronics Engineers (IEEE), pp. 35–42, Jun. 2009. doi:10.1109/mra.2009.932529.

[17] D. Debnath, F. Vanegas, J. Sandino, A. F. Hawary, and F. Gonzalez, "A Review of UAV Path-Planning Algorithms and Obstacle Avoidance Methods for Remote Sensing Applications," Remote Sensing, vol. 16, no. 21. MDPI AG, p. 4019, Oct. 29, 2024. doi:10.3390/rs16214019.

[18] W. Zhang, S. Zhang, F. Wu, and Y. Wang, "Path Planning of UAV Based on Improved Adaptive Grey Wolf Optimization Algorithm," IEEE Access, vol. 9. Institute of Electrical and Electronics Engineers (IEEE), pp. 89400–89411, 2021. doi:10.1109/access.2021.3090776.

[19] Z. Qadir, M. H. Zafar, S. K. R. Moosavi, K. N. Le, and M. A. P. Mahmud, "Autonomous UAV Path-Planning Optimization Using Metaheuristic Approach for Predisaster Assessment," IEEE Internet of Things Journal, vol. 9, no. 14. Institute of Electrical and Electronics Engineers (IEEE), pp. 12505–12514, Jul. 15, 2022. doi:10.1109/jiot.2021.3137331.

[20] L. Zhou, Z. Luo, P. Jiao, X. Jin, X. Sun, and J. Shi, “A Priority-Based Hybrid Algorithm for UAV Path Planning,” 2024 10th International Conference on Big Data and Information Analytics (BigDIA). IEEE, pp. 518–523, Oct. 25, 2024. doi:10.1109/bigdia63733.2024.10808417.
[21] Z. Zhang, J. Jiang, J. Wu, and X. Zhu, “Efficient and optimal penetration path planning for stealth unmanned aerial vehicle using minimal radar cross-section tactics and modified A-Star algorithm,” ISA Transactions, vol. 134. Elsevier BV, pp. 42–57, Mar. 2023. doi:10.1016/j.isatra.2022.07.032.
[22] R. Zhen, Q. Gu, Z. Shi, and Y. Suo, “An Improved A-Star Ship Path-Planning Algorithm Considering Current, Water Depth, and Traffic Separation Rules,” Journal of Marine Science and Engineering, vol. 11, no. 7. MDPI AG, p. 1439, Jul. 18, 2023. doi:10.3390/jmse11071439.
[23] K. Hao, Y. Yang, Z. Li, Y. Liu, and X. Zhao, “CERRT: A Mobile Robot Path Planning Algorithm Based on RRT in Complex Environments,” Applied Sciences, vol. 13, no. 17. MDPI AG, p. 9666, Aug. 26, 2023. doi:10.3390/app13179666.
[24] P. Xin, X. Wang, X. Liu, Y. Wang, Z. Zhai, and X. Ma, “Improved Bidirectional RRT* Algorithm for Robot Path Planning,” Sensors, vol. 23, no. 2. MDPI AG, p. 1041, Jan. 16, 2023. doi:10.3390/s23021041.
[25] W. Zhao, L. Li, Y. Wang, H. Zhan, Y. Fu, and Y. Song, “Research on A Global Path-Planning Algorithm for Unmanned Arial Vehicle Swarm in Three-Dimensional Space Based on Theta*–Artificial Potential Field Method,” Drones, vol. 8, no. 4. MDPI AG, p. 125, Mar. 27, 2024. doi:10.3390/drones8040125.
[26] Y. Chen, G. Luo, Y. Mei, J. Yu, and X. Su, “UAV path planning using artificial potential field method updated by optimal control theory,” International Journal of Systems Science, vol. 47, no. 6. Informa UK Limited, pp. 1407–1420, Jun. 18, 2014. doi:10.1080/00207721.2014.929191.
[27] K. Sreelakshmy, H. Gupta, I. A. Ansari, S. Sharma, K. K. Goyal, and O. P. Verma, “Metaheuristic Optimization for Three Dimensional Path Planning of UAV,” Lecture Notes in Networks and Systems. Springer Nature Singapore, pp. 791–802, 2022. doi:10.1007/978-981-19-0707-4_71.
[28] M. S. Adam, N. F. Abdullah, A. Abu-Samah, O. A. Amodu, and R. Nordin, “Advanced Path Planning for UAV Swarms in Smart City Disaster Scenarios Using Hybrid Metaheuristic Algorithms,” Drones, vol. 9, no. 1. MDPI AG, p. 64, Jan. 16, 2025. doi:10.3390/drones9010064.
[29] E. Baş and E. Ülker, “Improved social spider algorithm for large scale optimization,” Artificial Intelligence Review, vol. 54, no. 5. Springer Science and Business Media LLC, pp. 3539–3574, Nov. 09, 2020. doi:10.1007/s10462-020-09931-5.
[30] X. Yu, N. Jiang, X. Wang, and M. Li, “A hybrid algorithm based on grey wolf optimizer and differential evolution for UAV path planning,” Expert Systems with Applications, vol. 215. Elsevier BV, p. 119327, Apr. 2023. doi:10.1016/j.eswa.2022.119327.
[31] C. Zhong, G. Li, and Z. Meng, “Beluga whale optimization: A novel nature-inspired metaheuristic algorithm,” Knowledge-Based Systems, vol. 251. Elsevier BV, p. 109215, Sep. 2022. doi:10.1016/j.knosys.2022.109215.
[32] E. H. Houssein and A. Sayed, “Dynamic Candidate Solution Boosted Beluga Whale Optimization Algorithm for Biomedical Classification,” Mathematics, vol. 11, no. 3. MDPI AG, p. 707, Jan. 30, 2023. doi:10.3390/math11030707.
[33] S. Mirjalili, “SCA: A Sine Cosine Algorithm for solving optimization problems,” Knowledge-Based Systems, vol. 96. Elsevier BV, pp. 120–133, Mar. 2016. doi:10.1016/j.knosys.2015.12.022.
[34] S. Arora and S. Singh, “Butterfly optimization algorithm: a novel approach for global optimization,” Soft Computing, vol. 23, no. 3. Springer Science and Business Media LLC, pp. 715–734, Mar. 08, 2018. doi:10.1007/s00500-018-3102-4.
[35] S. Arora and S. Singh, “An improved butterfly optimization algorithm with chaos,” Journal of Intelligent & Fuzzy Systems, vol. 32, no. 1. SAGE Publications, pp. 1079–1088, Jan. 13, 2017. doi:10.3233/jifs-16798.
[36] A. Mortazavi and M. Moloodpoor, “Enhanced Butterfly Optimization Algorithm with a New fuzzy Regulator Strategy and Virtual Butterfly Concept,” Knowledge-Based Systems, vol. 228. Elsevier BV, p. 107291, Sep. 2021. doi:10.1016/j.knosys.2021.107291.
[37] M. D. Phung and Q. P. Ha, “Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization,” Applied Soft Computing, vol. 107. Elsevier BV, p. 107376, Aug. 2021. doi:10.1016/j.asoc.2021.107376.
[38] S. Mirjalili and A. Lewis, “The Whale Optimization Algorithm,” Advances in Engineering Software, vol. 95. Elsevier BV, pp. 51–67, May 2016. doi:10.1016/j.advengsoft.2016.01.008.
[39] R. Jiang, M. Yang, S. Wang, and T. Chao, “An improved whale optimization algorithm with armed force program and strategic adjustment,” Applied Mathematical Modelling, vol. 81. Elsevier BV, pp. 603–623, May 2020. doi:10.1016/j.apm.2020.01.002.
[40] Y. Huang, Y. Li, Z. Zhang, and Q. Sun, “A novel path planning approach for AUV based on improved whale optimization algorithm using segment learning and adaptive operator selection,” Ocean Engineering, vol. 280. Elsevier BV, p. 114591, Jul. 2023. doi:10.1016/j.oceaneng.2023.114591.
[41] W. Guo, T. Liu, F. Dai, and P. Xu, “An improved whale optimization algorithm for forecasting water resources demand,” Applied Soft Computing, vol. 86. Elsevier BV, p. 105925, Jan. 2020. doi:10.1016/j.asoc.2019.105925.
[42] M. Hooshyar and Y.-M. Huang, “Meta-heuristic Algorithms in UAV Path Planning Optimization: A Systematic Review (2018–2022),” Drones, vol. 7, no. 12, p. 687, Nov. 2023, doi: 10.3390/drones7120687.
[43] S. Agrawal, B. K. Patle, and S. Sanap, “A systematic review on metaheuristic approaches for autonomous path planning of unmanned aerial vehicles,” Drone Syst. Appl., vol. 12, pp. 1–28, Jan. 2024, doi: 10.1139/dsa-2023-0093.
[44] M. S. Adam, N. F. Abdullah, A. Abu-Samah, O. A. Amodu, and R. Nordin, “Advanced Path Planning for UAV Swarms in Smart City Disaster Scenarios Using Hybrid Metaheuristic Algorithms,” Drones, vol. 9, no. 1, p. 64, Jan. 2025, doi: 10.3390/drones9010064.
[45] B. Baidya, A. Mondal, S. Mallick, and S. Anamalamudi, “Energy-Efficient UAV Path Planning Using PSO-ABC Algorithm in Obstacle-Rich Environments,” 2024 OITS International Conference on Information Technology (OCIT). IEEE, pp. 199–204, Dec. 12, 2024. doi: 10.1109/ocit65031.2024.00043.
[46] G. Altun and İ. Aydın, “Optimizing Unmanned Vehicle Navigation: A Hybrid PSO-GWO Algorithm for Efficient Route Planning,” Firat University Journal of Experimental and Computational Engineering, vol. 4, no. 1, pp. 100–114, Feb. 2025, doi: 10.62520/fujece.1501508.
[47] N. Tenniche and B. Mendil, “A New Hybrid Water Cycle Algorithm for Path Planning with Obstacle Avoidance for Indoor Assistant Autonomous UAV Navigation,” Period. Polytech. Elec. Eng. Comp. Sci., vol. 69, no. 2, pp. 175–187, Apr. 2025, doi: 10.3311/ppee.39765.

[48] G. Hu, F. Huang, B. Shu, and G. Wei, “MAHACO: Multi-algorithm hybrid ant colony optimizer for 3D path planning of a group of UAVs,” Information Sciences, vol. 694, p. 121714, Mar. 2025, doi: 10.1016/j.ins.2024.121714.

[49] R. Özdağ, “A novel hybrid path planning method for sweep coverage of multiple UAVs,” J Supercomput, vol. 81, no. 1, Oct. 2024, doi: 10.1007/s11227-024-06574-z.

[50] Q. Shen, D. Zhang, M. Xie, and Q. He, “Multi-Strategy Enhanced Dung Beetle Optimizer and Its Application in Three-Dimensional UAV Path Planning,” Symmetry, vol. 15, no. 7, p. 1432, Jul. 2023, doi: 10.3390/sym15071432.

[51] E. Belge, A. Altan, and R. Hacıoğlu, “Metaheuristic Optimization-Based Path Planning and Tracking of Quadcopter for Payload Hold-Release Mission,” Electronics, vol. 11, no. 8, p. 1208, Apr. 2022, doi: 10.3390/electronics11081208.

[52] S. Aslan and T. Erkin, “DUALIPA: a new immune plasma algorithm for path planning of unmanned aerial vehicles,” Cluster Comput, vol. 28, no. 4, Feb. 2025, doi: 10.1007/s10586-024-04941-2.

[53] M. Wu et al., “Multi-strategy hybrid adaptive dung beetle optimization for UAV photogrammetric 3D path planning under complex constraints,” Sci Rep, vol. 15, no. 1, Apr. 2025, doi: 10.1038/s41598-025-98563-2.

[54] K. R. Opara and J. Arabas, “Differential Evolution: A survey of theoretical analyses,” Swarm and Evolutionary Computation, vol. 44. Elsevier BV, pp. 546–558, Feb. 2019. doi:10.1016/j.swevo.2018.06.010.

Authors’ Profiles

Mykola Nikolaiev: PhD student, Department of computer engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute” Kyiv, 03056, Ukraine.

Areas of scientific interests: Distributed systems, optimization methods, algorithms, machine learning.

Mykhailo Novotarskyi: Doctor of Sciences, Professor, Department of computer engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute” Kyiv, 03056, Ukraine.

Areas of scientific interests: Mathematical modeling of complex physical processes, formal description of mathematical and simulation models, parallel locally asynchronous numerical methods, machine learning and data mining, reinforcement learning methods.

Artem Volokyta: Candidate of Sciences, Associate Professor, Department of Computer Engineering, National Technical University of Ukraine, “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine.

Areas of scientific interests: High-performance computer systems and networks: theory, methods and means of hardware and software implementation; network topological organization.

How to cite this paper: Mykola Nikolaiev, Mykhailo Novotarskyi, Artem Volokyta, "A Novel Hybrid Differential Evolution and Enhanced Whale Optimization Algorithm for UAV Path Planning", International Journal of Information Technology and Computer Science(IJITCS), Vol.17, No.6, pp.109-126, 2025. DOI:10.5815/ijitcs.2025.06.06