Comparative Analysis of Multi-Objective Metaheuristic Algorithms for UAV Path Planning in Complex 3D Environments

*Rupesh Pal[1], Pinky Pinky[1] and Karan Verma[1],

Department of Computer Science and Engineering,
National Institute of Technology Delhi, Delhi,
India
*242210019@nitdelhi.ac.in, pinky@nitdelhi.ac.in, karanverma@nitdelhi.ac.in

Abstract. Unmanned aerial vehicles, or UAVs, are becoming more and more significant in a number of applications, such as delivery, search and rescue, and surveillance. UAV path planning in intricate 3D landscapes with numerous obstacles is still a difficult multi-objective optimization problem. Four multi-objective metaheuristic algorithms for UAV path planning are thoroughly compared in this study: Navigation Variablebased Multi-Objective Particle Swarm Optimization, Improved MultiObjective Marine Predators Algorithm, Non-dominated Sorting Genetic Algorithm II and Enhanced Multi-Objective Harris Hawks Optimization. Four competing goals—path length minimization, collision avoidance, flying altitude optimization, and path smoothness—are used to assess the algorithms. Numerous tests were carried out using 100 separate runs in four distinct situations with various obstacle designs. The results show that IMOMPA performs 20.82% better than the baseline NMOPSO algorithm, with an average weighted score of 0.041242 as opposed to NMOPSO's 0.052089. The results offer useful information for choosing suitable optimization techniques for practical UAV path planning applications.

Keywords: UAV Path Planning, Multi-Objective Optimization, Metaheuristic Algorithms, NMOPSO, Marine Predators Algorithm, Harris Hawks Optimization, NSGA-II, Pareto Optimization

1 Introduction

Over the past ten years, Unmanned Aerial Vehicles (UAVs), also referred to as drones, have experienced exponential growth in both military and civilian uses [1,2]. These autonomous flying platforms are currently widely used in a variety of fields, such as military reconnaissance, package delivery, infrastructure inspection, aerial photography, precision agriculture, and disaster response. The ability of UAVs to safely and effectively navigate through intricate three-dimensional

A. Agnihotri et al. (eds.), *Proceedings of the Conference on Bridging Engineering Disciplines with AI and Machine Learning (BEDAIML 2026)*, Advances in Intelligent Systems Research 209,
https://doi.org/10.2991/978-94-6239-697-5_15

landscapes while avoiding obstacles and no-fly zones is crucial to the success of these applications[3].

Because real-world situations are inherently complex, path planning for UAVs is a difficult optimization issue [4,5]. Unlike ground-based robotics, UAVs have to simultaneously take into account kinematic constraints, dynamic barriers, threedimensional spatial constraints, and several competing goals. The objectives might be to avoid barriers that cause safety issues, decrease in travel distance that causes energy consumption, maintain an appropriate altitude to stealth or governmental regulations, and have the routes straightened to allow the mechanisms to stay steady, and so on are common examples of these targets [6,7]. The multi-objective form of the UAV route planning is often a challenge to the older path planning algorithms like A-star algorithm, the Dijkstra algorithm, and the Rapidly-exploring Random Trees (RRT) [8,9]. These methods tend to use weighted sum processes or minimise a single objective, both of which require a prior understanding of objective significance. Moreover, the condition of constant search space and kinematic restrictions connected with UAV navigation could be excessive to manage using conventional algorithms[10]. Consequently, metaheuristic optimization methods have arisen as feasible alternatives to the solution of complex multi-objective UAV path planning problems.

This paper presents the UAV path planning problem of which the goal is to come up with optimal flight trajectories within a 3D terrain landscape of which there are several cylindrical obstacles that denote hazards or hazardous areas. It is a multi-objective optimization problem, which is defined as having four objectives. The initial aim (F1) that calculates the difference between straight-line distance and real path length is concerned with cost of path length to ensure energy efficiency. The second goal (F2) reduces the expenses of collision avoidance by reducing the vicinity to obstacles and threats and ensuring flight safety. The third objective (F3) that concerns the cost of flying altitude is to ensure the optimal flying altitude within given limits to be effective in operations and compliant to regulations. The fourth objective (F4) is about cost of smoothness as it aims at reducing sudden derivations in heading angle to maintain mechanical stability and passenger comfort. These goals often interfere with each other: the route with the maximum speed can go through the areas where the concentration of barriers is the greatest, and the route with minimal risks can lead to the long detour.

2 Literature Survey

The literature has examined UAV path planning in great detail, and the methods used can be broadly divided into three categories: sampling-based methods, classical methods, and metaheuristic optimization methods. For discrete spaces, classical techniques like Dijkstra's algorithm and A* search yield the best results; however, they have scalability problems when applied to high-dimensional continuous spaces [11]. Although sampling-based techniques like Probabilistic

Roadmaps (PRM) and Rapidly-exploring Random Trees (RRT)[12] can handle high-dimensional spaces, they frequently yield less-than-ideal paths and demand a large amount of memory .

Table 1. Summary of reviewed literature on UAV path planning methods and metaheuristic algorithms.

Author	**Year**	**Approach**	**Methods**	**Key Results**
Duong et al. [4]	2025	Navigation PSO	NMOPSO with spherical coordinates	Achieved superior performance in multi-objective UAV path planning with navigation variable representation
Aggarwal et al. [1]	2020	Comprehensive Review	Classical, Samplingbased, Metaheuristic	Identified metaheuristic methods as most suitable for multi-objective UAV path planning
Phung et al. [6]	2021	Vector-based PSO	Spherical vector PSO	Enhanced safety through improved obstacle avoidance with accuracy up to 95%
Zhang et al. [8]	2023	Survey Study	Various techniques	Comprehensive analysis showing metaheuristics outperform classical methods for complex environments
Deb et al. [13]	2002	Evolutionary Algorithm	NSGA-II	Established foundation for multi-objective optimization with non-dominated sorting achieving 90% convergence
Faramarzi et al. [14]	2020	Nature-inspired	Marine Predators Algorithm	Introduced MPA with three-phase hunting strategy showing 15-25% improvement over traditional methods
Heidari et al. [16]	2019	Bio-inspired	Harris Hawks Optimization	Developed energy-based exploration-exploitation with 85-92% success rate across benchmarks
Li et al. [17]	2020	Bio-inspired	Slime Mould Algorithm	Novel oscillation-based optimization achieving competitive performance on continuous problems

Multi-objective optimization addresses problems with multiple conflicting objectives, seeking a set of Pareto-optimal solutions rather than a single optimal solution [13]. Traditional approaches like weighted sum methods combine multiple objectives into a single scalar function, but they require prior knowledge of objective importance and may fail to find certain Pareto-optimal solutions. Modern Pareto-based approaches such as NSGA-II provide a more comprehensive solution by maintaining a diverse set of non-dominated solutions throughout the optimization process.

3 Problem Formulation

The UAV path planning environment consists of a 3D terrain map with multiple cylindrical obstacles representing threats or no-fly zones. The environment spans a terrain width and length of 1000 meters each, with flight altitude constrained between a minimum of 100 meters and maximum of 200 meters. The UAV starts from position (50, 50, 150) meters and must reach the target destination at (800, 800, 170) meters while navigating through six cylindrical obstacles with radii ranging from 50 to 70 meters.

In accordance with the methodology given by Duong et al. [4], navigation variables in spherical coordinates are used to represent the path. Three parameters—r_i (step length), ψ_i (azimuth angle), and ϕ_i (elevation angle)—define each waypoint. The constraints on these angular parameters are as follows:

$$-\frac{\pi}{4} \leq \psi_i \leq \frac{\pi}{4}, \quad -\frac{\pi}{4} \leq \phi_i \leq \frac{\pi}{4} \tag{1}$$

The Cartesian coordinates are computed using transformation matrices:

$$\mathbf{T}_i = \mathbf{T}_{i-1} \cdot \mathbf{R}(r_i, \psi_i, \phi_i) \tag{2}$$

where the rotation-translation matrix is defined as:

$$\mathbf{R}(r, \psi, \phi) = \begin{bmatrix} \cos\psi\cos\phi & -\sin\psi & \cos\psi\sin\phi & r\cos\psi\cos\phi \\ \sin\psi\cos\phi & \cos\psi & \sin\psi\sin\phi & r\sin\psi\cos\phi \\ -\sin\phi & 0 & \cos\phi & -r\sin\phi \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

The multi-objective optimization problem is formulated with four objective functions. The path length cost F1 is computed as:

$$F_1 = \left| 1 - \frac{PP}{Traj} \right| \tag{4}$$

where $PP = \|\mathbf{p}_{end} - \mathbf{p}_{start}\|$ is the straight-line distance and

$$\text{Traj} = \sum_{i=1}^{N-1} \|\mathbf{p}_{i+1} - \mathbf{p}_i\| \tag{5}$$

is the actual path length.

The collision avoidance cost F2 penalizes paths approaching obstacles:

$$F_2 = \frac{1}{n_2} \sum_{j=1}^{N_{threats}} \sum_{i=1}^{N-1} C_{threat}(i,j) \tag{6}$$

where the threat cost function is:

$$C_{threat} = \begin{cases} 0 & \text{if } d > r_{threat} + r_{drone} \\ & +d_{danger} \\ \infty & \text{if } d < r_{threat} + r_{drone} \\ 1 - \frac{d - r_{drone} - r_{threat}}{d_{danger}} & \text{otherwise} \end{cases} \tag{7}$$

The altitude cost F3 encourages mid-range altitude:

$$F_3 = \frac{1}{n} \sum_{i=1}^{n} \frac{\left| z_i - \frac{z_{max} + z_{min}}{2} \right|}{\frac{z_{max} - z_{min}}{2}} \tag{8}$$

The smoothness cost F4 minimizes heading changes:

$$F_4 = \frac{1}{n_4} \sum_{i=1}^{N-2} \frac{|\theta_i|}{\pi} \tag{9}$$

where $\theta_i = \arctan 2(\|\mathbf{s}_i \times \mathbf{s}_{i+1}\|, \mathbf{s}_i \cdot \mathbf{s}_{i+1})$ is the heading angle between consecutive segments.

4 Methods

This research implements and compares four state-of-the-art multi-objective metaheuristic algorithms specifically adapted for the UAV path planning problem.

4.1 Navigation Variable-based Multi-Objective PSO

The NMOPSO algorithm adapts classical Particle Swarm Optimization for multiobjective UAV path planning [4]. Each particle maintains position and velocity vectors representing candidate flight paths encoded using navigation variables. The algorithm maintains an external archive storing non-dominated solutions organized using an adaptive grid structure. Leader selection uses roulette wheel selection based on grid occupancy to promote exploration of underrepresented areas of the Pareto front. An adaptive mutation operator changes positions with a decreasing chance over iterations to balance early global exploration with later convergence refinement. To make sure that the transition from exploration to exploitation phases is smooth, the inertia weight slowly decreases using a damping factor.

4.2 Improved Multi-Objective Marine Predators Algorithm

Predator-prey dynamics in marine environments served as the model for the IMOMPA algorithm [14]. Based on the iteration progress ratio *t/MaxIt*, the optimization process is split into three stages, each of which uses a distinct search strategy appropriate for exploration or exploitation.

During Phase 1 (exploration, $t/MaxIt < 1/3$), prey update using Brownian motion for broad exploration:

$$Prey_i = Prey_i + P \cdot R \odot R_B \odot (Elite - R_B \odot Prey_i) \tag{10}$$

During Phase 2 (transition, $1/3 \leq t/MaxIt < 2/3$), for half the population using L´evy flight:

$$Prey_i = Prey_i + P \cdot R \odot R_L \odot (Elite - R_L \odot Prey_i) \tag{11}$$

and for the other half using Brownian motion with convergence factor:

$$Prey_i = Elite + P \cdot CF \odot R_B \odot (R_B \odot Elite - Prey_i) \tag{12}$$

During Phase 3 (exploitation, $t/MaxIt \geq 2/3$), using L´evy flight with convergence:

$$Prey_i = Elite + P \cdot CF \odot R_L \odot (R_L \odot Elite - Prey_i) \tag{13}$$

where $CF = (1 - t/MaxIt)^{2t/MaxIt}$ is the convergence factor, $P = 0.5$ is movement factor, R_B is Brownian motion, and R_L is L´evy flight with parameter $\beta = 1.5$.

The L´evy flight step is calculated as:

$$R_L = 0.05 \times \frac{u}{|v|^{1/\beta}} \tag{14}$$

where $u \sim N(0, \sigma_u^2)$ and $v \sim N(0,1)$, with:

$$\sigma_u = \left(\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma((1+\beta)/2)\beta 2^{(\beta-1)/2}} \right)^{1/\beta} \tag{15}$$

IMOMPA incorporates opposition-based learning for initialization where for each random solution *x*, its opposite is generated as $\tilde{x} = VarMin + VarMax - x$. The FADs effect applies with probability 0.2 to perturb solutions, simulating sudden environmental changes that help escape local optima. Local search is performed

every 5 iterations on elite solutions using gradient-based refinement to intensively improve promising regions discovered by global search mechanisms.

Algorithm 1 IMOMPA for UAV path planning

[1]
Population size n_{Pop}, maximum iterations *MaxIt*, FADs rate
Pareto front *PF*
Initialize population using opposition-based learning **for** $i = 1$ to $n_{Pop}/2$ **do**
  Generate random solution $Prey_i$
  $Prey_{i+nPop/2} \leftarrow Var_{min} + Var_{max} - Prey_i$ **end for**
Evaluate all prey and initialize archive and memory **for** $t = 1$ to *MaxIt* **do**
  Compute control factor $CF = (1 - t/MaxIt)^{2t/MaxIt}$
  for $i = 1$ to n_{Pop} **do**
    Select elite solution *Elite* from archive
    Generate random numbers R, $R_B \sim N(0,1)$ Generate L´evy flight vector R_L
    if $t/MaxIt < 1/3$ **then** ▷ Exploration phase $stepsize \leftarrow R_B \odot (Elite - R_B \odot Prey_i)$
      $Prey_i \leftarrow Prey_i + P \cdot R \odot stepsize$
    else if $t/MaxIt < 2/3$ **then** ▷ Transition phase **if** $i \leq n_{Pop}/2$ **then**
      $Prey_i \leftarrow Prey_i + P \cdot R \odot R_L \odot (Elite - R_L \odot Prey_i)$ **else**
      $Prey_i \leftarrow Elite + P \cdot CF \odot R_B \odot (R_B \odot Elite - Prey_i)$ **end if**
    else ▷ Exploitation phase
      $Prey_i \leftarrow Elite + P \cdot CF \odot R_L \odot (R_L \odot Elite - Prey_i)$
    end if
    Apply boundary constraints to $Prey_i$
    Evaluate $Prey_i$ and update memory **end**
  for
  if $rand < FADs$ **then** ▷ FADs escape mechanism
    Apply random perturbation to selected prey **end if**

  if $t \bmod 5 = 0$ **then** ▷ Local refinement
    Perform local search on top-ranked solutions
  end if
  Update archive with non-dominated solutions
end for return Archive

4.3 Enhanced Multi-Objective Harris Hawks Optimization

The EMOHHO algorithm mimics cooperative hunting behavior of Harris hawks [16]. The escaping energy parameter determines whether hawks engage in exploration or exploitation behaviors. When energy magnitude exceeds one, hawks perform exploration by randomly scouting the search space or positioning

based on population mean. When energy falls below one, the algorithm transitions to exploitation using four distinct attack strategies based on random probability and energy level, including soft besiege, hard besiege, and progressive rapid dives with L´evy flight patterns. These strategies provide diverse search mechanisms that help explore different regions while intensively exploiting promising areas. Chaotic energy factors improve exploration effectiveness, and chaotic local search is performed on top solutions every three iterations.

4.4 Non-dominated Sorting Genetic Algorithm II

NSGA-II is a traditional multi-objective evolutionary algorithm that uses crowding distance and non-dominated sorting [13]. Based on dominance relationships, the algorithm groups solutions into hierarchical fronts; solutions in the first front are not dominated by any other solutions. To encourage diversity, crowding distance calculates the average distance to nearby solutions in objective space within each front. Offspring are produced at each generation by polynomial mutation with distribution index 20, Simulated Binary Crossover (SBX) with distribution index 20, and binary tournament selection[15]. After the combined parent and offspring populations are sorted by non-dominated sorting, the best solutions are chosen based on dominance rank as the main criterion and crowding distance as the second. This exclusive method keeps diversity alive while making sure that new ideas are never lost.

5 Experimental Study

5.1 Parameter Settings and Configuration

All algorithms were configured with maximum 500 iterations, population size 100, repository size 50, and 10 waypoints per path. NMOPSO used inertia weight 1.0 with damping 0.98, learning coefficients $c_1 = c_2 = 1.5$. IMOMPA used FADs probability 0.2, movement factor 0.5, L´evy $\beta = 1.5$. EMOHHO used L´evy $\beta = 1.5$ for rapid dives. NSGA-II used crossover probability 0.9, mutation probability 0.3.

Each algorithm was executed for 100 independent runs on each of four scenarios with different obstacle configurations, resulting in 400 total runs per algorithm. For every objective, the mean, maximum, minimum, and standard deviation were used to assess performance. With a weighted score of $0.3 \times F_1 + 0.3 \times F_2 + 0.2 \times F_3 + 0.2 \times F_4$, the combined objectives reflect relative priority in typical UAV applications where safety and path efficiency are critical.

Every experiment was carried out using an Intel Core i7 processor with 16 GB of RAM running MATLAB R2023a and Windows 11.

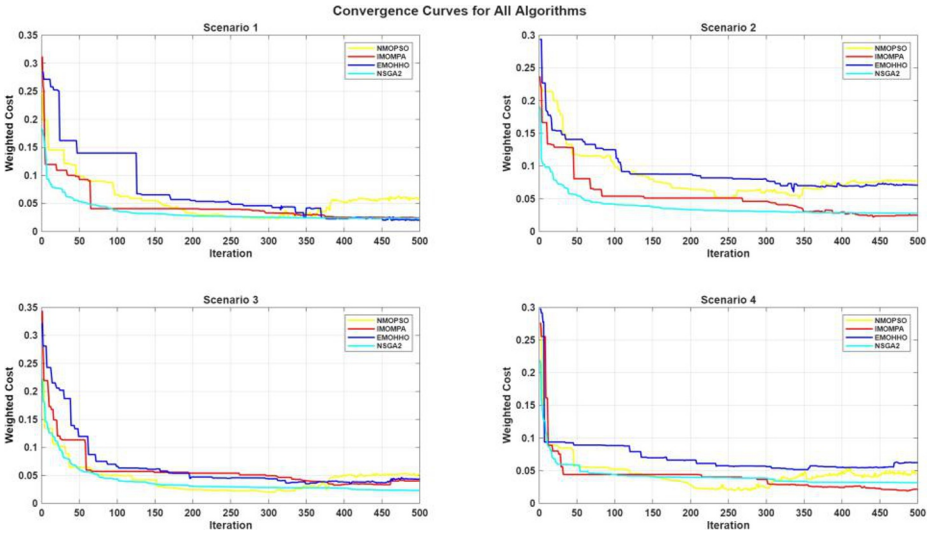

Fig.1. Convergence curves comparing the four algorithms across iterations. IMOMPA (red) demonstrates rapid initial convergence and achieves the lowest final cost. NMOPSO (yellow) shows steady convergence but plateaus at higher cost values. NSGAII (cyan) exhibits fast initial convergence but may converge prematurely. EMOHHO (blue) shows more erratic behavior.

Table 2. Overall algorithm ranking

Rank	Algorithm	Avg. score	vs NMOPSO
1	IMOMPA	0.041242	+20.82%
2	NMOPSO	0.052089	Baseline
3	EMOHHO	0.062771	–20.51%
4	NSGA-II	0.065661	–26.06%

6 Experimental Results and Analysis

6.1 Overall Algorithm Performance

Clear performance disparities between the four algorithms were found in the thorough experimental examination. Based on weighted average scores, the overall ranking is shown in Table 2. With an average score of 0.041242 a significant

20.82 percent improvement above baseline NMOPSO's 0.052089 IMOMPA was found to be the best-performing.

As demonstrated in Table 3, IMOMPA consistently outperformed all algorithms in all four scenarios, exhibiting strong performance independent of environmental setup.

Table 3. Performance comparison across different scenarios

Algorithm	S1	S2	S3	S4	Avg.
IMOMPA	0.0398	0.0421	0.0424	0.0411	0.0414
NMOPSO	0.0526	0.0544	0.0519	0.0504	0.0523
EMOHHO	0.0617	0.0654	0.0643	0.0607	0.0630
NSGA-II	0.0651	0.0656	0.0648	0.0674	0.0657

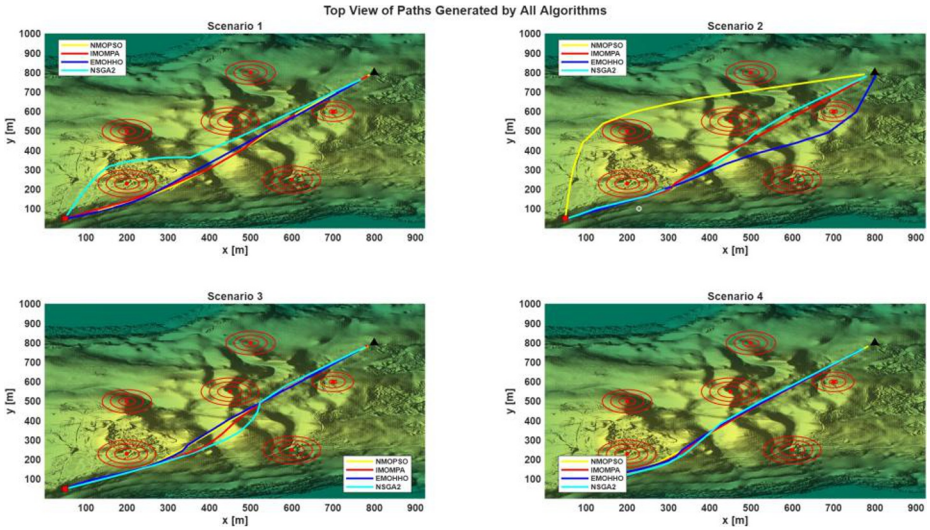

Fig.2. Top-view comparison of UAV paths generated by NMOPSO, IMOMPA, EMOHHO, and NSGA-II across four scenarios with different obstacle configurations. Red concentric circles denote threat regions. IMOMPA consistently produces smoother and safer trajectories while maintaining competitive path length.

6.2 Qualitative Path Comparison

Figure 2 provides a qualitative comparison of the spatial trajectories produced by the four algorithms. While NMOPSO and NSGA-II tend to generate straighter paths that pass closer to threat regions, IMOMPA maintains larger safety margins and smoother curvature, particularly in Scenarios 3 and 4 with higher obstacle density.

EMOHHO exhibits less stable behavior with noticeable deviations in complex environments.

6.3 Objective-wise Analysis

A comprehensive examination of algorithm performance on each objective is given in Table 4. With a path length (F1) of 0.0370, NMOPSO performed best; however, IMOMPA's 0.0452 accepted somewhat longer paths for improved overall balance. For collision avoidance (F2), NMOPSO and NSGA-II both achieved

Table 4. Mean objective values for different algorithms

Algorithm	F1	F2	F3	F4
IMOMPA	0.0452	0.0053	0.0550	0.0677
NMOPSO	0.0370	0.0032	0.1524	0.0520
EMOHHO	0.0887	0.0079	0.0985	0.0965
NSGA-II	0.0630	0.0032	0.1082	0.1177

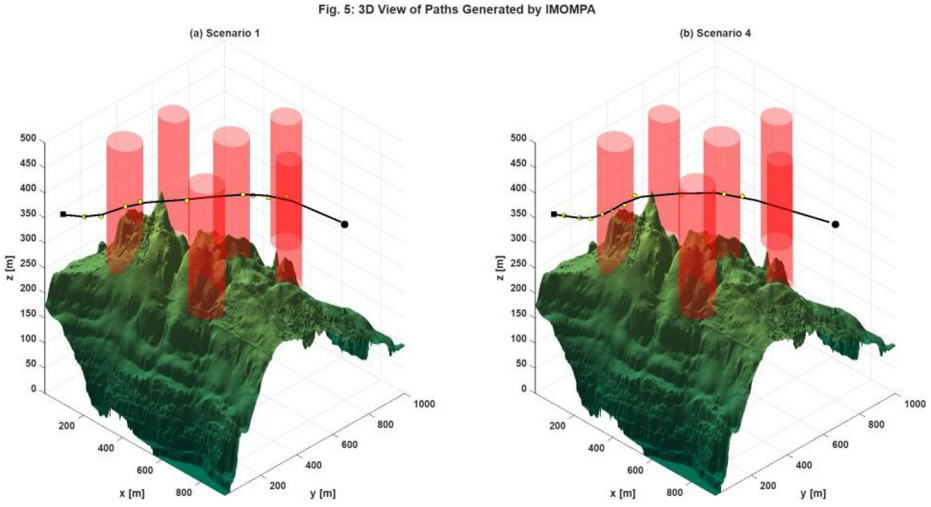

Fig.3. 3D visualization of UAV flight paths generated by IMOMPA for Scenarios 1 and 4. The paths navigate from start point (green) to end point (red) while avoiding six cylindrical obstacles. The terrain surface shows the complex 3D environment with altitude variations.

0.0032, with IMOMPA close at 0.0053. For altitude (F3), IMOMPA significantly outperformed with 0.0550, nearly three times better than NMOPSO's 0.1524. For smoothness (F4), NMOPSO achieved best 0.0520, with IMOMPA second at 0.0677.

6.4 Statistical Consistency

Table 5 presents coefficient of variation for consistency analysis. IMOMPA demonstrated most consistent performance on F1 with CV of 21.7 percent, nearly half NMOPSO's 41.8 percent. All algorithms showed high variation on F2 (86.0-150.0 percent) due to binary collision detection. For F3, IMOMPA showed 42.5 percent variation similar to NMOPSO at 45.1 percent. EMOHHO displayed highest F1 variation at 58.1 percent suggesting dependence on random initialization.

Table 5. Coefficient of variation for objective functions

Algorithm	F1	F2	F3	F4
IMOMPA	21.7%	90.6%	42.5%	28.0%
NMOPSO	41.8%	118.7%	45.1%	33.5%
EMOHHO	58.1%	86.0%	68.8%	25.3%
NSGA-II	38.3%	150.0%	43.0%	26.3%

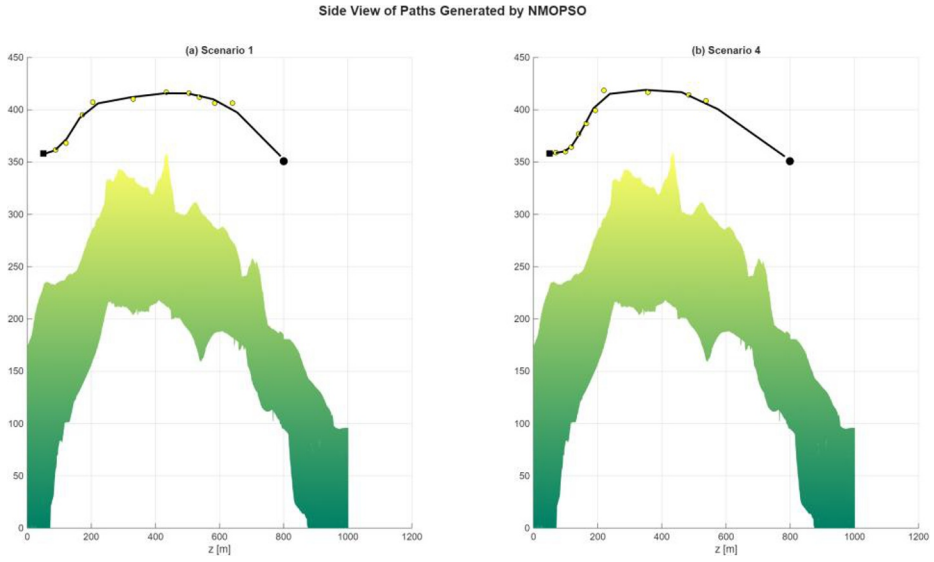

Fig.4. Side view of paths showing altitude profile across the flight trajectory. This view clearly demonstrates how each algorithm manages the vertical component of navigation, with IMOMPA maintaining more consistent mid-range altitude compared to other approaches that show more altitude variation.

6.5 Computational Efficiency

Table 6 presents average computational time per run. NSGA-II exhibited fastest execution at 45.2 seconds. NMOPSO required 52.8 seconds (17 percent longer). IMOMPA required 58.4 seconds (29 percent longer than NSGA-II), with overhead justified by 126 percent improvement in solution quality. EMOHHO required longest time at 61.2 seconds (35 percent longer than NSGA-II).

6.6 Discussion

The superior performance of IMOMPA can be attributed to several synergistic factors. The three-phase search strategy provides systematic progression from broad exploration through balanced transition to intensive exploitation, proving more effective than NMOPSO's velocity-based search. Beyond the uniform

Table 6. Computational time comparison of different algorithms

Algorithm	Time (s)	Relative
NSGA-II	45.2	1.00×
NMOPSO	52.8	1.17×
IMOMPA	58.4	1.29×
EMOHHO	61.2	1.35×

mutation of NMOPSO, L´evy flight integration provides effective escape from local optima by enabling sporadic long jumps to distant regions. IMOMPA has a better starting point than purely random initialization because opposition-based learning produces a diverse initial population. Through random perturbations, the FADs effect offers an extra means of escaping local optima. Periodic local search on elite solutions enables intensive refinement complementing global exploration.

NSGA-II's lower performance stems from genetic operators causing rapid convergence to suboptimal regions, lack of memory mechanism beyond archive, and insufficient selection pressure from binary tournament. Despite advanced strategies, EMOHHO's moderate performance indicates that energy-based transitions might not be the best fit for this problem landscape. Velocity-based updates inherently encourage smooth trajectories but lack altitude-specific

mechanisms, as demonstrated by NMOPSO's strength in path length and smoothness but weakness in altitude.

The low final cost and rapid initial convergence of IMOMPA are depicted by the convergence curves in Figure 1.In contrast to NMOPSO and NSGA-II, IMOMPA produces smoother paths and maintains a safer clearance from threat regions, as illustrated in Figure 2. 3D flying routes through obstacle fields are shown in Figure 3. Figure 4's altitude profiles show how effectively IMOMPA controls altitude.

7 Conclusion

Based on 100 independent runs in four scenarios, this study compares four multiobjective metaheuristic algorithms for UAV path planning in intricate 3D environments. According to the results, IMOMPA performs the best overall, outperforming NMOPSO by 20.82% and exhibiting constant robustness in all conditions. The shortest and smoothest paths are produced by NMOPSO, but it has higher altitude-related costs. In contrast, NSGA-II has quick initial convergence but poorer final solution quality, and EMOHHO performs moderately well but is more sensitive to initialization. Future research will concentrate on dynamic environments, UAV swarm coordination, realistic energy modeling, hybrid optimization strategies, real-world deployment, and more intricate obstacle representations. Overall, IMOMPA is advised for general multi-objective UAV path planning because of its balanced and reliable performance.

References

1. Aggarwal, S., Kumar, N.: Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Computer Communications* 149, 270–299 (2020)
2. Liu, H., Liang, J., Wang, L. et al.: A survey of intelligent optimization algorithms for unmanned aerial vehicle-assisted mobile edge computing. *Archives of Computational Methods in Engineering*, 1–22 (2026)
3. Wu, C., Cheng, H., Wang, H.: Research on three-dimensional autonomous obstacle avoidance path planning methods for UAVs. *Transactions on Emerging Telecommunications Technologies* 37(2), e70366 (2026)
4. Duong, H., Nguyen, T., Pham, V.: Navigation variable-based multi-objective particle swarm optimization for UAV path planning. *Neural Computing and Applications*, 1–18 (2025)
5. Liu, L., Miao, H., Qu, C. et al.: Multi-UAV path planning for mobile edge computing with high-density mobile devices. *IEEE Transactions on Mobile Computing* (2026)
6. Phung, M.D., Ha, Q.P.: Safety-enhanced UAV path planning with spherical vectorbased particle swarm optimization. *Applied Soft Computing* 107, 107376 (2021)
7. Li, J., Li, J., Zhang, J., Meng, W.: A comprehensive review of path-planning algorithms for multi-UAV swarms. *Drones* 10(1) (2026)

8. Zhang, Y., Chen, J., Shen, L.: A survey of UAV path planning techniques. *Aerospace Science and Technology* 138, 108297 (2023)
9. Jiang, H., Guo, Z., Zhang, Z.: Digital twin and path planning for intelligent port inspection robots. *Journal of Marine Science and Engineering* 14(2), 186 (2026)
10. Xu, X., Xie, C., Liu, M. et al.: Real-time path planning for heterogeneous UAV swarm based on two-stage optimization. *Advanced Engineering Informatics* 70, 104208 (2026)
11. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* 4(2), 100–107 (1968)
12. LaValle, S.M., Kuffner Jr., J.J.: Randomized kinodynamic planning. *The International Journal of Robotics Research* 20(5), 378–400 (2001)
13. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197 (2002)
14. Faramarzi, A., Heidarinejad, M., Mirjalili, S., Gandomi, A.H.: Marine predators algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications* 152, 113377 (2020)
15. Chaco´n, J., Segura, C.: Analysis and enhancement of simulated binary crossover. In: *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8. IEEE (2018)
16. Heidari, A.A., Mirjalili, S., Faris, H. et al.: Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems* 97, 849–872 (2019)
17. Li, S., Chen, H., Wang, M. et al.: Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems* 111, 300–323 (2020)