Proceedings of CONF-MPCS 2025 Symposium: Mastering Optimization: Strategies for Maximum Efficiency
DOI: 10.54254/2753-8818/2025.CH22587

Research on UAV Path Planning Strategy Based on Particle Swarm Optimization Algorithm: Performance Analysis and Optimization of Inertial Weights and Mutated Particles

Haoyang Shen

Institute of Information Science and Technology, Southwest Jiaotong University, Chengdu, China
shy1254001934@my.swjtu.edu.cn

Abstract: Unmanned Aerial Vehicle (UAV) path planning is a critical challenge in autonomous navigation, requiring intelligent trajectory generation in complex environments that balance safety, efficiency, and adaptability. Particle Swarm Optimization (PSO), a prominent swarm intelligence algorithm, is widely used for its robust global search capabilities. However, PSO's performance heavily depends on parameter configurations, particularly the inertia weight (ω). This paper investigates the impact of parameter settings on PSO-based UAV path planning and proposes optimization strategies to enhance its efficacy.Simulation experiments reveal that when ω = 0.5, PSO exhibits strong local search capabilities and fast convergence but is prone to local optima, reducing path planning precision. Increasing ω enhances global search, enabling broader solution space exploration, but at the cost of slower convergence and limited optimization gains due to computational constraints. Notably, when ω = 1.5, the algorithm avoids local optima but shows minimal performance improvement and reduced efficiency.The study also evaluates optimization strategies, including a linearly varying inertia weight and a mutant particle strategy. The linearly varying inertia weight significantly improves optimization performance but slows convergence, making it suitable for high-accuracy scenarios with ample computational resources. The mutant particle strategy, however, achieves a better balance between optimization and convergence, ideal for practical applications requiring both efficiency and accuracy. Combining these strategies does not yield substantial improvements, suggesting that a single strategy may be more computationally efficient.This research provides valuable insights into PSO parameter configuration and optimization for UAV path planning, advancing the field of autonomous navigation technology.

Keywords: UAV path planning, particle swarm optimization (PSO), inertia weight, optimization strategies, global and local search

1. Introduction

In the context of rapid technological advancement, UAV [1] technology has made significant progress and is now widely applied in fields such as military, agriculture, logistics, and disaster monitoring. UAV path planning, as a critical component of autonomous navigation, aims to determine a safe and efficient flight trajectory [2-3] that meets mission requirements. However, due to complex environmental constraints, dynamic obstacles, and the need for multi-objective optimization, this

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2753-8818/2025.CH22587

planning problem is highly nonlinear and complex, and traditional methods often fail to achieve the desired results [4]. In recent years, path planning methods based on intelligent optimization algorithms have garnered considerable attention for their efficiency and robustness. Among these methods, Particle Swarm Optimization (PSO) has emerged as a research hotspot because of its simplicity, efficiency, and strong global search capabilities [5].

PSO is a swarm intelligence-based optimization algorithm that simulates the social behavior observed in bird flocking or fish schooling, leveraging both individual and collective search mechanisms to locate optimal solutions [6]. In UAV path planning, PSO effectively addresses complex constraints and quickly finds near-optimal trajectories. However, its performance is heavily reliant on parameter settings, especially the selection of the inertia weight (ω) [7], which determines the exploration and exploitation capabilities of particles in the search process and directly influences the algorithm's global search capacity and convergence speed [8]. Consequently, studying the effects of different parameter settings on PSO's performance is of great significance for enhancing the efficiency and accuracy of UAV path planning.

This paper systematically investigates the influence of various parameter settings on UAV path planning performance based on the PSO algorithm [9-10]. Initially, a series of simulation experiments are conducted to analyze how inertia weight affects the optimization performance and convergence speed of PSO [11]. The experimental findings indicate that when ω = 0.5, the algorithm demonstrates robust local search capability and the fastest convergence, though it tends to get trapped in local optima. As ω increases, the algorithm's global search ability is enhanced, allowing it to explore a wider solution space; however, the convergence speed decreases markedly, and the extent of performance improvement is constrained by computational resources and search space dimensions. The paper also explores the application effects of different PSO variants—such as linearly varying inertia weight and mutant particle strategies—in UAV path planning. Results show that while the linearly varying inertia weight significantly enhances optimization performance (albeit with slower convergence), it is better suited for scenarios requiring high planning accuracy with sufficient computational resources. The mutant particle strategy, on the other hand, strikes a good balance between optimization performance and convergence speed, making it preferable for practical applications that demand both efficiency and accuracy [12-13].

2. Particle swarm optimization algorithm

Particle Swarm Optimization (PSO) is an algorithm modeled after the foraging behavior observed in birds. In nature, birds operate in groups to collectively identify the most efficient route to food. As this process unfolds, each bird tracks its own path as personal experience. Meanwhile, every bird in the flock contributes its individual experiences, which are shared across the group, forming a shared knowledge base. Through repeated foraging, both individual birds and the group continuously assess and refine their understanding of the optimal path. This constant updating of personal and global best paths brings the flock closer to the ideal solution with each iteration, slowly converging toward the true optimum.

The fundamental concept of PSO translates individual birds into particles and the flock into a swarm of particles. As these particles move through the search space, each one adjusts its velocity and direction based on its own experience as well as the collective experience of the swarm. This iterative adjustment propels the swarm toward an optimal solution, eventually leading to convergence on the best possible outcome.

The velocity update formula in one-dimensional space is given by:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 [pBest_i(t) - pos_i(t)] + c_2 r_2 [gBest_i(t) - pos_i(t)] \quad (1)$$

DOI: 10.54254/2753-8818/2025.CH22587

where: $v_i(t)$ represents the velocity of particle i at time t. ω is the inertia weight, which controls the particle's ability to maintain its original velocity. c_1 is the cognitive parameter, indicating the influence of the particle's own experience. $c_2$is the social parameter, representing the influence of the group's best experience. r_1 and r_2 are random numbers uniformly distributed between 0 and 1, i.e., $r1$,$r2 \in [0,1]$. $pBest_i(t)$ is the historical best position of particle i. $gBest_i(t)$ represents the global best position of the particle swarm.

The position update formula in one-dimensional space is given by:

$$pos_i(t+1) = pos_i(t) + v_i(t+1) \tag{2}$$

where $pos_i(t)$ represents the position of particle i at time t, and dt represents the time step.

In the context of three-dimensional path planning, the path of each particle from the starting point to the endpoint can be considered as a series of three-dimensional points. Therefore, the position of a particle can be represented as:

$$pos_i = \{(x_{i1}, y_{i1}, z_{i1}), (x_{i2}, y_{i2}, z_{i2}), \dots, (x_{in}, y_{in}, z_{in})\} \tag{3}$$

Similarly, the velocity of a particle can be represented as:

$$v_i = \{(v_{x_{i1}}, v_{y_{i1}}, v_{z_{i1}}), (v_{x_{i2}}, v_{y_{i2}}, v_{z_{i2}}), \dots, (v_{x_{in}}, v_{y_{in}}, v_{z_{in}})\} \tag{4}$$

The following sections extend the discussion from one-dimensional to three-dimensional space regarding the velocity and position update formulas.

Velocity update formula in three-dimensional space is given by:

$$\begin{cases} v_{x_{ij}}(t+1) & = \omega v_{x_{ij}}(t) + c_1 r_1 \left[pBest_{x_{ij}}(t) - x_{ij}(t)\right] + c_2 r_2 \left[gBest_{x_{ij}}(t) - x_{ij}(t)\right] \\ v_{y_{ij}}(t+1) & = \omega v_{y_{ij}}(t) + c_1 r_1 \left[pBest_{y_{ij}}(t) - y_{ij}(t)\right] + c_2 r_2 \left[gBest_{y_{ij}}(t) - y_{ij}(t)\right] \\ v_{z_{ij}}(t+1) & = \omega v_{z_{ij}}(t) + c_1 r_1 \left[pBest_{z_{ij}}(t) - z_{ij}(t)\right] + c_2 r_2 \left[gBest_{z_{ij}}(t) - z_{ij}(t)\right] \end{cases} \tag{5}$$

Position update formula in three-dimensional space is given by:

$$\begin{cases} x_{ij}(t+1) = x_{ij}(t) + v_{x_{ij}}(t+1) \\ y_{ij}(t+1) = y_{ij}(t) + v_{y_{ij}}(t+1) \\ z_{ij}(t+1) = z_{ij}(t) + v_{z_{ij}}(t+1) \end{cases} \tag{6}$$

Limitations on velocity and position is given by:

$$v_{x_{ij}} \in [v_{x_{min}}, v_{x_{max}}],\ v_{y_{ij}} \in [v_{y_{min}}, v_{y_{max}}],\ v_{z_{ij}} \in [v_{z_{min}}, v_{z_{max}}] \tag{7}$$

$$x_{ij} \in [x_{min}, x_{max}],\ y_{ij} \in [y_{min}, y_{max}],\ z_{ij} \in [z_{min}, z_{max}] \tag{8}$$

Let the starting point be denoted as "startPos" and the target point as "goalPos". The fitness function is defined as:

$$fitness(pos_i) = PathLength(pos_i) + Penalty(pos_i) \tag{9}$$

where: $PathLength(pos_i)$ is a function that calculates the length of the path represented by particle i. $Penalty(pos_i)$ is a penalty function used to calculate the distance between the particle and obstacles. If the particle intersects with an obstacle, the penalty value is increased significantly to make the fitness value large.

DOI: 10.54254/2753-8818/2025.CH22587

In each generation, the fitness of all particles is compared, and the best one is identified to update the global best position. The goal is to minimize the fitness function. The formula for updating the global best position is:

$$gBest(t+1) = min\{fitness(pos_i)\} \tag{10}$$

To sum up, the logic diagram of the PSO based UAV path planning algorithm is shown in Figure1, and Figure 2 is the schematic diagram of the simulation results

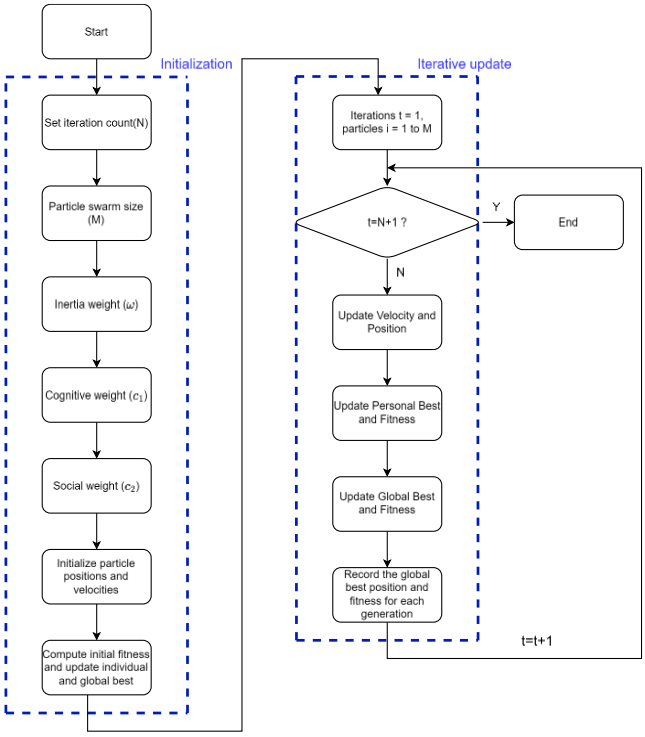

Figure 1: Algorithm flowchart

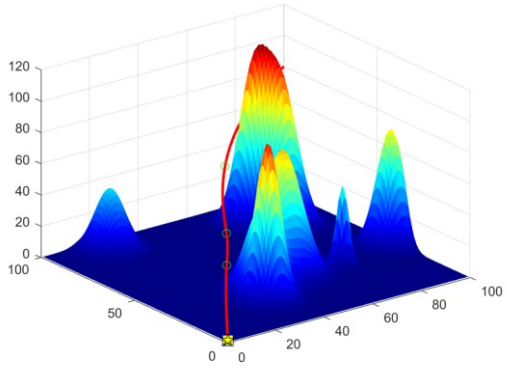

Figure 2: Simulation result diagram

DOI: 10.54254/2753-8818/2025.CH22587

3. Simulation test under different working conditions

3.1. The average fitness corresponding to different ω from 0.5 to 1.5

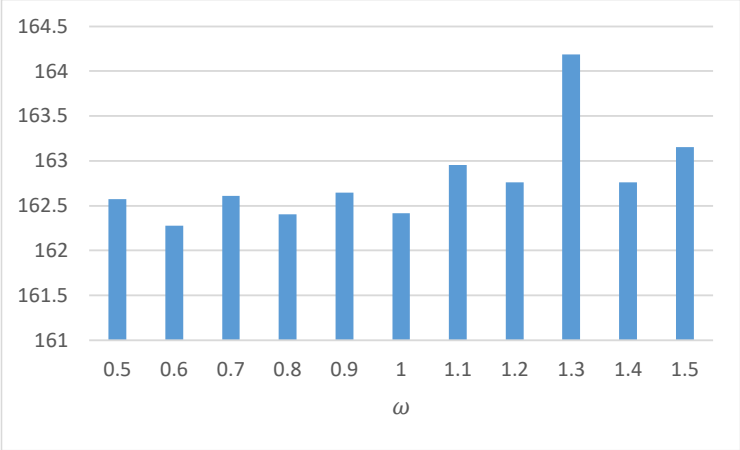

Figure 3: Average fitness

Figure 3 shows the average fitness corresponding to different ω: The lowest fitness occurs when ω = 0.6. In this case, the particle swarm optimization (PSO) algorithm demonstrates strong local search ability with relatively high search accuracy. The algorithm is able to find good solutions in a short amount of time; however, its global search capability is weaker. As ω increases from 0.5 to 1.5, although theoretically the PSO algorithm should be more likely to escape local optima and perform global searches, the simulation results indicate that the algorithm does not find significantly better solutions. Instead, the fitness remains at a slightly higher level. This suggests that although increasing the inertia weight may help avoid local optima, in actual simulations, the particle swarm fails to fully identify the global optimum, likely due to an excessively large search space or computational resource limitations. Therefore, the fitness does not significantly decrease.

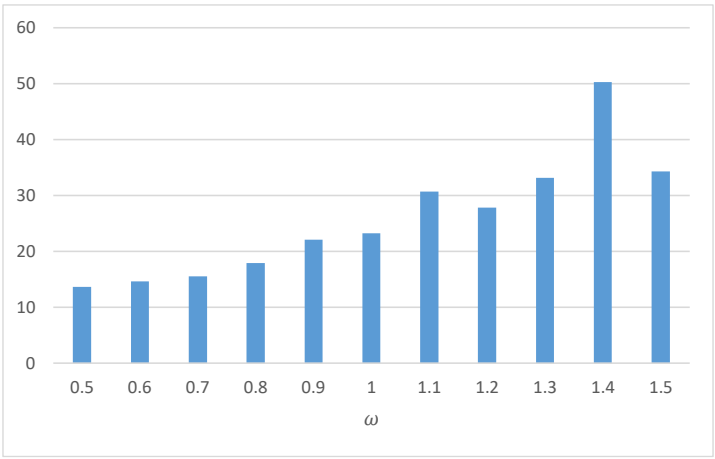

Figure 4: Average iterations

Figure 4 shows the average fitness iterations to different ω:At ω=0.5, the convergence speed is the fastest, requiring the fewest iterations. When the inertia weight ω is smaller, the PSO algorithm converges faster and is able to find feasible solutions with fewer iterations. A smaller ω value means the particle swarm relies more on local search, thus accelerating the convergence process. However, the cost of rapid convergence is a higher tendency to fall into local optima. As the inertia weight

DOI: 10.54254/2753-8818/2025.CH22587

increases, the convergence speed of the PSO algorithm slows down. Increasing the inertia weight enhances the particle's exploration ability and helps prevent premature convergence to local optima. However, this also results in more iterations, as the particle swarm may search a larger range and needs more iterations to converge.

Comprehensive Performance Evaluation:

From the comprehensive analysis of optimization performance and convergence speed, the setting of ω=0.6 appears to exhibit a better balance, avoiding early convergence to local optima while ensuring higher optimization performance. However, as ω increases further, while the global search capability of the algorithm improves, the fitness does not increase significantly, and the convergence speed slows down. This indicates that increasing the inertia weight does not unconditionally enhance performance. Therefore, the inertia weight in PSO should be adjusted appropriately based on the specific optimization problem and objectives. Optimization performance should consider both search efficiency and solution accuracy.

Future Research Directions:

PSO has shown good performance in many complex optimization problems but still faces challenges such as local optima traps and slow convergence speed. Future research could explore hybrid methods that combine PSO with other optimization algorithms, such as genetic algorithms, ant colony optimization, or simulated annealing. These algorithms each have different strengths and weaknesses. By combining the advantages of different algorithms, it may be possible to overcome the limitations of a single algorithm and improve overall optimization performance and efficiency.

3.2. Comparison of different algorithms under fixed map conditions

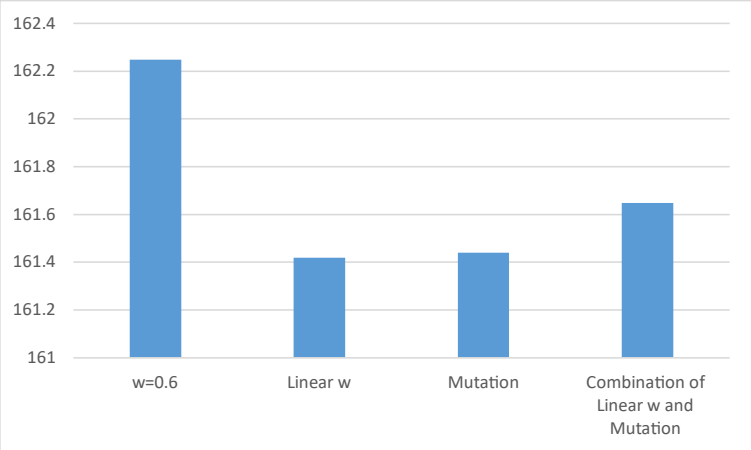

Figure 5: Average fitness

Figure 5 shows the average fitness of different algorithms after multiple tests on a fixed map:In the scenario where the number and shape of peaks in a fixed map are altered, experimental results indicate that the lowest fitness value occurs when the inertia weight changes linearly (161.4183), suggesting that this strategy offers the best optimization performance. By dynamically adjusting the inertia weight, this approach effectively balances global and local search capabilities, overcoming the problem of premature convergence.

The fitness value of the method incorporating mutation particles ranks second (161.4398), indicating that the introduction of the mutation mechanism can enhance the global search ability of the algorithm to some extent, though it is slightly less effective than the linear change of the inertia weight.

DOI: 10.54254/2753-8818/2025.CH22587

The fitness value of combining mutation particles with inertia weight (161.6475) is better than that of fixed inertia weight but inferior to the first two strategies, indicating that the improvement effect is limited.

The fitness value for fixed inertia weight is the highest (162.2472), but its optimization effect is the worst, suggesting that the lack of a dynamic adjustment mechanism leads to a higher likelihood of falling into local optima.

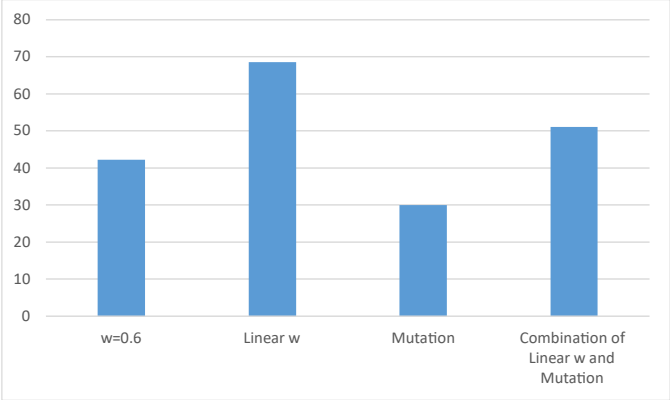

Figure 6: Average iterations

Figure 6 shows the average iterations of different algorithms after multiple tests on a fixed map.The approach incorporating mutation particles requires the fewest average iterations (30) and exhibits the fastest convergence speed while demonstrating relatively good optimization performance. This indicates that it achieves a good balance between convergence speed and optimization effectiveness.

The fixed inertia weight approach converges relatively quickly (42.17) but shows the worst optimization performance, implying that it tends to rapidly converge to local optima.

The combination of mutation particles and inertia weight results in a slower convergence speed (51) and does not yield better optimization results than the mutation particle-only approach, indicating that the improvement strategy has not significantly enhanced performance.

The strategy with linearly changing inertia weight has the slowest convergence speed (68.5), but it produces the best optimization performance, suggesting that it sacrifices some convergence speed in favor of tronger global search capabilities.

3.3. Comparison of different algorithms under random maps conditions

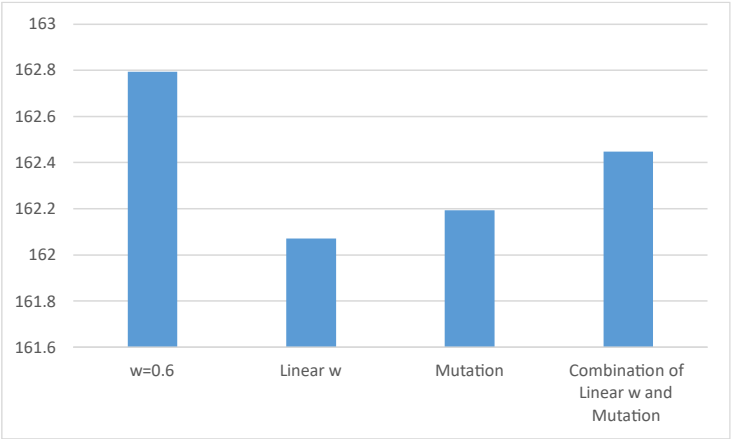

Figure 7: Average fitness

DOI: 10.54254/2753-8818/2025.CH22587

Figure 7 shows the average fitness. of different algorithms after multiple tests on random maps. The lowest fitness value (162.0719) occurs when the inertia weight changes linearly, indicating the best optimization performance. This strategy, by dynamically adjusting the inertia weight, effectively balances global and local search capabilities, preventing the algorithm from prematurely converging to local optima.

The method that incorporates mutation particles ranks second in fitness value (162.194), suggesting that the introduction of the mutation mechanism can, to some extent, enhance the global search ability of the algorithm. However, it is slightly less effective than the linear change in inertia weight.

The fitness value of the combination of mutation particles and inertia weight (162.4477) is better than that of a fixed inertia weight but inferior to the first two strategies, indicating that the improvement effect is limited.

The fitness value for fixed inertia weight is the highest (162.7935), but its optimization effect is the worst. This indicates that the absence of a dynamic adjustment mechanism leads to a higher tendency for the algorithm to fall into local optima.

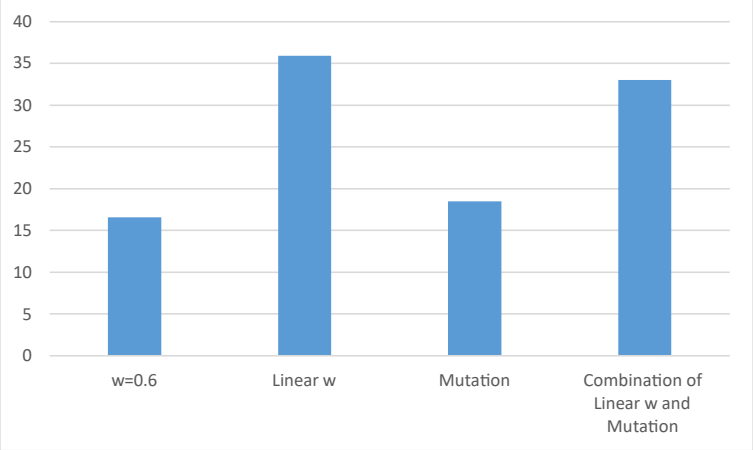

Figure 8: Average iterations

Figure 8 shows the average iterations of different algorithms after multiple tests on random maps. The fixed inertia weight approach requires the fewest average iterations (16.6) and converges the fastest. However, its optimization performance is poor, indicating that it tends to quickly converge to local optima.

The approach with added mutation particles converges relatively quickly (18.5) while demonstrating better optimization performance. This suggests that it strikes a good balance between convergence speed and optimization effectiveness.

The combination of mutation particles and inertia weight results in slower convergence (33) and inferior optimization performance compared to the method with mutation particles alone. This indicates that the improvement strategy has not significantly enhanced performance.

The strategy with linearly changing inertia weight converges the slowest (35.9), but it yields the best optimization performance. This suggests that, by sacrificing some convergence speed, it achieves stronger global search capabilities.

Comprehensive Performance Evaluation:

The strategy with linearly changing inertia weight exhibits the best optimization performance but has a slower convergence speed. It is suitable for applications that require high path-planning accuracy and sufficient computational resources.

DOI: 10.54254/2753-8818/2025.CH22587

The addition of mutation particles achieves a good balance between optimization performance and convergence speed, making it a more optimal strategy for scenarios that need to balance planning accuracy with computational efficiency.

The combination of mutation particles and inertia weight yields limited improvement and does not significantly enhance the algorithm's performance. Therefore, it is not recommended to prioritize this strategy in practical applications.

Fixed inertia weight, although converging the fastest, yields the poorest optimization performance. It is only suitable for scenarios where planning accuracy is not critical, and rapid solutions are required.

Future Research Directions: Further exploration of more complex dynamic inertia weight adjustment strategies (e.g., nonlinear adjustments) could enhance the algorithm's optimization performance. Optimization of key parameters in the algorithm, such as inertia weight and mutation probability, may improve the algorithm's robustness and overall performance. Parallel computing techniques can be leveraged to accelerate algorithm execution, reduce computational costs, and enhancethe algorithm's practicality.

4. Conclusion

This paper systematically investigates the impact of parameter settings and algorithm variants on UAV path planning using Particle Swarm Optimization (PSO). Key findings reveal that the inertia weight (ω) significantly influences PSO performance. At ω = 0.5, PSO demonstrates strong local search capabilities and fast convergence but is prone to local optima, limiting path planning precision. Increasing ω enhances global search, enabling broader solution space exploration, but slows convergence and offers limited optimization gains due to computational constraints. Notably, at ω = 1.5, the algorithm avoids local optima but shows minimal performance improvement and reduced efficiency.

The study also explores optimization strategies, including a linearly varying inertia weight and a mutant particle strategy. The linearly varying inertia weight improves optimization performance but slows convergence, making it suitable for high-precision scenarios with ample computational resources. The mutant particle strategy, however, achieves a better balance between optimization and convergence, ideal for practical applications requiring both efficiency and accuracy. Combining these strategies does not yield significant improvements, suggesting that a single strategy may be more computationally efficient.

Despite its potential, PSO faces challenges such as local optima and slow convergence, particularly in complex environments. Future research should focus on adaptive parameter mechanisms, hybrid algorithms combining PSO with other optimization methods, multi-objective approaches considering factors like path length and energy consumption, and real-time planning in dynamic environments. This research provides valuable insights for optimizing PSO in UAV path planning, advancing autonomous navigation technology.

References

[1] *Tjiharjadi S, Razali S, Sulaiman H A. A systematic literature review of multi-agent pathfinding for maze research[J]. Journal of Advances in Information Technology Vol, 2022, 13(4).*

[2] *Zahaf, A., Abboudi, A., Bounemeur, A., Bououden, S., & Chemachema, M. (2024). A Constrained PSO Based Model Predictive Control for Mobile Robot System with Sensor Faults. Lecture Notes in Electrical Engineering, 1147 LNEE, 333–342.*

[3] *Mao-Hui F, Jun X. Research on trajectory planning algorithm of unmanned aerial vehicle based on improved A* algorithm[C]//2017 International Conference on Computer Systems, Electronics and Control (ICCSEC). IEEE, 2017: 1348-1352.*

DOI: 10.54254/2753-8818/2025.CH22587

[4] *Zhang, L., Liu, J., & Wang, Z. (2021). Adaptive particle swarm optimization for UAV path planning in dynamic en vironments. Journal of Intelligent & Robotic Systems, 102(1), 1-18. https://doi.org/10.1007/s10846-021-01430-9*

[5] *Jusof M F M, Mohammad S, Razak A A A, Hybrid Manta ray foraging—Particle swarm algorithm for PD control optimization of an inverted pendulum[C]//Recent Trends in Mechatronics Towards Industry 4.0: Selected Articles from iM3F 2020, Malaysia. Springer Singapore, 2022: 1-13.*

[6] *Lu Y, Zhang X. Multiple Unmanned Aerial Vehicles Path Planning Based on Collaborative Differential Evolution[C]//International Conference on Swarm Intelligence. Cham: Springer Nature Switzerland, 2023: 98-110.*

[7] *Li, Y., Chen, H., & Wang, X. (2022). Enhanced PSO-based path planning for UAVs in complex environments. IEEE Transactions on Aerospace and Electronic Systems, 58(3), 2105-2118. https://doi.org/10.1109/TAES.2022.3156789*

[8] *Wang, Y., Zhang, X., & Li, J. (2023). Multi-objective PSO for UAV path planning with obstacle avoidance and energy efficiency. Applied Soft Computing, 132, 109856. https://doi.org/10.1016/j.asoc.2023.109856*

[9] *Chen, X., Li, Q., & Zhang, Y. (2022). A hybrid PSO-GA algorithm for UAV path planning in 3D environments. Aerospace Systems, 5(2), 123-136. https://doi.org/10.1007/s42401-022-00129-9*

[10] *Koh J S, Tan R H G, Lim W H, A modified particle swarm optimization for efficient maximum power point tracking under partial shading condition[J]. IEEE Transactions on Sustainable Energy, 2023.*

[11] *Li J, Xiong X, Yan Y, A Survey of Indoor UAV Obstacle Avoidance Research[J]. IEEE Access, 2023.*

[12] *Yang Y, Xiong X, Yan Y. UAV Formation Trajectory Planning Algorithms: A Review[J]. Drones, 2023, 7(1): 62.*

[13] *Hsieh, F.-S. (2024). A Hybridized Scheme for Solving Ridesharing Problems Based on Firefly Algorithm and a Variant of PSO Algorithm. International Conference on Enterprise Information Systems, ICEIS - Proceedings, 2, 769–774.*