Proceedings of CONF-MSS 2025 Symposium: Automation and Smart Technologies in Petroleum Engineering
DOI: 10.54254/2755-2721/2025.KL22614

Optimizing Swarm Size in UAV Path Planning: Balancing Fitness, Convergence, and Computational Efficiency

Yuhao Su

School of Chemical Engineering and Technology, Tianjin University, Tianjin, China
su_yuhao@tju.edu.cn

Abstract: Unmanned aerial vehicle (UAV) path planning represents a pivotal component in autonomous navigation systems, with its primary objective being the determination of safe, efficient, and optimized flight trajectories within complex operational environments. Among various optimization methodologies, particle swarm optimization (PSO) has emerged as a prominent solution for path planning challenges, owing to its computational efficiency and straightforward implementation. Nevertheless, the efficacy of PSO is intrinsically linked to its parameter configuration, where different parameter selections can substantially influence the algorithm's search capacity and convergence dynamics. This research systematically examines the effects of four critical PSO parameters—inertia weight, social weight, cognitive weight, and swarm size—on both optimization quality and computational performance in UAV path planning scenarios. The investigation commences with the formulation of a mathematical framework for UAV path planning, incorporating path length and obstacle avoidance capability as primary evaluation metrics. Subsequently, through comprehensive simulation experiments, we meticulously analyze the influence of various parameter configurations on PSO's convergence characteristics, employing advanced data fitting techniques to establish precise mathematical relationships between parameter settings and algorithmic performance. The empirical results reveal that strategic parameter optimization substantially enhances PSO's global search capability, improves path quality, and expedites convergence rates. Based on rigorous analysis, the optimal parameter configuration is determined to be: inertia weight (w) = 0.659, social weight (c_1) = 1.349, cognitive weight (c_2) = 2.108, and swarm size (M) = 88. These findings not only provide theoretical foundations but also offer practical optimization strategies for parameter selection in UAV path planning applications, with potential extensions to other intelligent optimization domains.

Keywords: UAV path planning, particle swarm optimization, parameter tuning, inertia weight, social weight

1. Introduction

With the increasing application of UAVs in intelligent transportation, logistics, environmental monitoring, and disaster relief, path planning has become a crucial research focus in autonomous flight [1]. The fundamental goal of UAV path planning is to determine an optimal flight trajectory that ensures safety while meeting operational requirements, improving efficiency, and minimizing energy consumption. However, due to environmental uncertainties, complex obstacle distributions,

© 2026 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: 10.54254/2755-2721/2025.KL22614

and dynamic mission constraints, traditional path planning methods struggle in high-dimensional spaces, often suffering from high computational complexity and susceptibility to local optima [2-3].

Particle swarm optimization (PSO) is a swarm intelligence-based global optimization method widely used in UAV path planning due to its simple structure, fast convergence, and strong adaptability. However, the performance of PSO is highly dependent on parameter selection [4]. The key parameters influencing convergence and optimization quality include inertia weight, social weight, cognitive weight, and swarm size. Inertia weight controls particle motion trends and balances global exploration with local convergence. Social weight determines the degree of reliance on the global best solution, influencing the stability of convergence [5]. Cognitive weight dictates how strongly a particle is influenced by its own best-known solution, affecting local optimization capability. Swarm size affects search space coverage and computational efficiency.

Most PSO applications use fixed parameter settings, which often fail to achieve an optimal balance between search ability and computational efficiency in different task scenarios. Inappropriate parameter choices may cause premature convergence, slow optimization, or inefficient exploration [6]. To enhance PSO's performance in UAV path planning, this study systematically analyzes the impact of these key parameters on algorithmic performance through simulation experiments. Using data fitting methods, we establish mathematical models describing the relationships between parameters and optimization performance, ultimately determining an optimal parameter set that maximizes PSO's efficiency in path planning. The findings of this research provide parameter tuning strategies and theoretical support for UAV path planning and other intelligent optimization problems [7-8].

2. Particle Swarm Optimization (PSO)

2.1. Basic principles

Particle Swarm Optimization (PSO) is an algorithm inspired by the foraging behavior of birds. In nature, birds collaborate in groups to find the most efficient path to food. During this process, each bird records its own path as personal experience [9]. At the same time, every member of the flock shares its individual experiences, which propagate throughout the group, contributing to a collective knowledge base. Through repeated foraging, both individual birds and the group as a whole evaluate and refine their understanding of the best path. This continuous updating of personal and global optimal paths drives the flock closer to the ideal solution with each iteration, gradually converging toward the true optimum. The core concept of PSO abstracts individual birds as particles and the flock as a swarm of particles. Through ongoing movement, each particle adjusts its velocity and direction based on both its own experience and the collective experience of the swarm. This iterative process propels the swarm toward the optimal solution, ultimately leading to convergence on the best possible outcome [10-11].

2.2. Process of PSO for solving the optimal path

The core equations of PSO include the velocity update formula and the position update formula. For a particle swarm consisting of N particles, at time T, they are represented as formulas [12] (1), (2):

$$\mathrm{v}_i(t+1) = w \cdot \mathrm{v}_i(t) + c_1 \cdot r_1 \cdot \left(\mathrm{pBest}_i(t) - \mathrm{Pos}_i(t)\right) + c_2 \cdot r_2 \cdot \left(\mathrm{gBest}(t) - \mathrm{Pos}_i(t)\right) \quad (1)$$

$$\mathrm{Pos}_i(t+1) = \mathrm{Pos}_i(t) + \mathrm{v}_i(t+1) \quad (2)$$

Where:$\mathrm{v}_i(t)$ is the velocity of particle i at time t,w is the inertia weight, controlling the influence of the particle's previous velocity on the next moment,c_1 is the cognitive parameter influenced by the particle's personal experience, c_2 is the cognitive parameter influenced by the swarm's

DOI: 10.54254/2755-2721/2025.KL22614

experience,r_1 r_2 are random numbers uniformly distributed in [0, 1],$\mathrm{pBest}_i(t)$ is the individual best position of particle i at time t,$\mathrm{gBest}(t)$ is the global best position of the swarm at time t,$\mathrm{Pos_i(t)}$ is the position of particle i at time t

For solving the optimal path in a three-dimensional space, each particle represents a path, which is composed of n three-dimensional points. Thus, the particle's position and velocity can be represented as follows [13]:

$$\mathrm{Pos}_i(t) = \{(x_{i1}, y_{i1}, z_{i1}), (x_{i2}, y_{i2}, z_{i2}) \dots (x_{in}, y_{in}, z_{in})\} \tag{3}$$

$$\mathrm{v}_i(t) = \{(v_{xi1}, v_{yi1}, v_{zi1})\} \tag{4}$$

To apply PSO in a three-dimensional space, the velocity update formula needs to be expanded for each dimension. The velocity update formula is as follows:

$$\begin{cases} \mathrm{v}_{xij}(t+1) = w \cdot \mathrm{v}_{xij}(t) + c_1 \cdot r_1 \cdot \left(\mathrm{pBest}_{xij}(t) - \mathrm{x}_{ij}(t)\right) + c_2 \cdot r_2 \cdot \left(\mathrm{gBest}_{xij}(t) - \mathrm{x}_{ij}(t)\right) \\ \mathrm{v}_{yij}(t+1) = w \cdot \mathrm{v}_{yij}(t) + c_1 \cdot r_1 \cdot \left(\mathrm{pBest}_{yij}(t) - \mathrm{y}_{ij}(t)\right) + c_2 \cdot r_2 \cdot \left(\mathrm{gBest}_{yij}(t) - \mathrm{y}_{ij}(t)\right) \\ \mathrm{v}_{zij}(t+1) = w \cdot \mathrm{v}_{zij}(t) + c_1 \cdot r_1 \cdot \left(\mathrm{pBest}_{zij}(t) - \mathrm{z}_{ij}(t)\right) + c_2 \cdot r_2 \cdot \left(\mathrm{gBest}_{xij}(t) - \mathrm{z}_{ij}(t)\right) \end{cases} \tag{5}$$

The position update formula is as follows:

$$\begin{cases} \mathrm{x}_{ij}(t+1) = \mathrm{x}_{ij}(t) + \mathrm{v}_{xij} \\ \mathrm{y}_{ij}(t+1) = \mathrm{y}_{ij}(t) + \mathrm{v}_{yij} \\ \mathrm{z}_{ij}(t+1) = \mathrm{z}_{ij}(t) + \mathrm{v}_{zij} \end{cases} \tag{6}$$

Considering these constraints, the velocity update formula is as follows:

$$\mathrm{v}_i(t+1) = w \cdot \mathrm{v}_i(t) + c_1 \cdot r_1 \cdot \left(\mathrm{pBest}_i(t) - \mathrm{Pos}_i(t)\right) + c_2 \cdot r_2 \cdot \left(\mathrm{gBest}(t) - \mathrm{Pos}_i(t)\right) \tag{7}$$

$$if\ \mathrm{v}_i \in [\mathrm{v}_{min}, \mathrm{v}_{max}]$$

$$\begin{cases} \mathrm{v}_i(t+1) = \mathrm{v}_{min}, & if\ \mathrm{v}_i < \mathrm{v}_{min} \\ \mathrm{v}_i(t+1) = \mathrm{v}_{max}, & if\ \mathrm{v}_i > \mathrm{v}_{max} \end{cases}$$

The position update formula is:

$$\begin{cases} \mathrm{Pos}_i(t+1) = \mathrm{Pos}_i(t) + \mathrm{v}_i(t+1), if\ \mathrm{Pos}_i \in [\mathrm{Pos}_{min}, \mathrm{Pos}_{max}] \\ \mathrm{Pos}_i(t+1) = \mathrm{Pos}_{min},\ if\ \mathrm{Pos}_i < \mathrm{Pos}_{min} \\ \mathrm{Pos}_i(t+1) = \mathrm{Pos}_{max},\ if\ \mathrm{Pos}_i > \mathrm{Pos}_{max} \end{cases} \tag{8}$$

For evaluating the quality of each particle's corresponding path, we introduce a fitness function. In flight path planning problems, the evaluation criteria are the path length and safety, i.e., whether there is any collision with obstacles. For a path from the start point to the end point, the fitness function is represented as follows [14]:

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

Where:

$fitness(Pos_i)$ is the fitness function of particle i,$PathLength(Pos_i)$ is the length of the path corresponding to particle i,$Penalty(Pos_i)$ is the penalty value when the path intersects with obstacles

Thus, we can find the global optimum, i.e., when i=k, $fitness(Pos_i)$ reaches the minimum value:

$$gBest(t+1) = Pos_k(t) \tag{10}$$

DOI: 10.54254/2755-2721/2025.KL22614

The specific flowchart is as Figure 1:

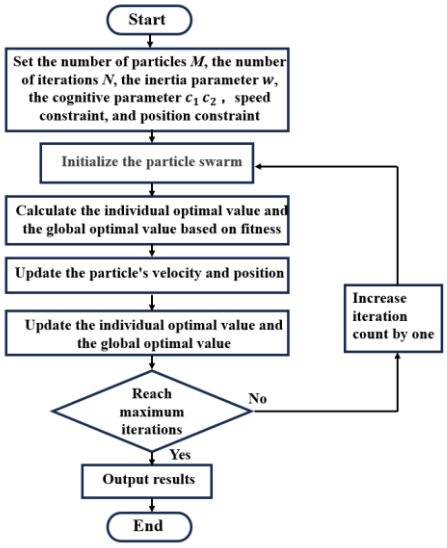

Figure 1: Flowchart of particle swarm optimization for solving the optimal path

3. The impact of inertia weight on PSO algorithm performance

The Particle Swarm Optimization (PSO) algorithm is a swarm intelligence-based optimization technique whose performance heavily depends on parameter tuning. Inertia weight (w) is a crucial parameter in PSO, controlling the velocity update of particles in the search space. It significantly affects the algorithm's global exploration and local exploitation abilities. An excessively large inertia weight may cause particles to oscillate within the search space, reducing convergence stability, while an excessively small inertia weight may limit the particles' exploration capability, leading to premature convergence [15].

This chapter analyzes the impact of varying inertia weight (w) from 0.5 to 1.5 on PSO performance in UAV path planning. To quantitatively assess the relationship between inertia weight and algorithm performance, this chapter conduct a curve fitting analysis and derive mathematical models that describe the relationship between inertia weight and both best fitness and number of iterations. Based on this analysis, we determine an optimal inertia weight that balances solution quality and convergence efficiency.

3.1. Curve fitting analysis

To quantify the impact of inertia weight on PSO performance, we established mathematical models describing the relationships between inertia weight (w) and best fitness and between inertia weight (w) and the number of iterations as Table 1. A quadratic polynomial fitting approach was used to analyze the experimental data as Figure 2.

Table 1: Relationship between inertia weight (w), best fitness, and number of iterations

w	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5
The number of iterations	12.3	15.7	13.6	11.8	16.9	19.8	21.8	27	25.6	37.9	26.9
Best fitness	162.7491	161.5941	162.7739	161.8220	161.7708	161.9495	162.2088	163.4853	162.4011	163.0628	162.7574

DOI: 10.54254/2755-2721/2025.KL22614



Figure 2: Relationship between inertia weight (w), best fitness, and number of iterations

3.1.1. Fitting of inertia weight and best fitness

Best fitness measures the quality of the final solution obtained by PSO; a lower best fitness value indicates better optimization results in path planning. Through experimental data analysis, the relationship between best fitness and inertia weight (w) was found to follow a quadratic polynomial function, as given by:

$$f(w) = 2.1222w^2 - 3.4660w + 163.5475 \tag{11}$$

The fitting results indicate that when w = 0.817, the best fitness reaches its minimum, meaning that the algorithm achieves the best optimization performance. However, when w further increases, the best fitness begins to rise, suggesting that excessive inertia weight may degrade solution quality by causing the particles to overshoot the optimal solution. Therefore, from an optimization quality perspective, the inertia weight should be close to 0.817 to ensure the best path planning performance.

3.1.2. Fitting of inertia weight and number of iterations

The number of iterations represents the convergence speed of the PSO algorithm; a smaller number of iterations means the algorithm finds the optimal solution more quickly, improving computational efficiency. Experimental results indicate that inertia weight (w) and number of iterations also follow a quadratic polynomial relationship, described by:

$$f(w) = 10.6410w^2 - 0.0911w + 9.2315 \tag{12}$$

From the fitted curve, we observe that when w = 0.5, the number of iterations reaches its minimum, indicating that the PSO algorithm converges the fastest. However, as w increases, the number of iterations rises significantly, suggesting that a larger inertia weight slows down convergence because particles maintain high velocities for prolonged periods, leading to inefficient searching near the optimal solution. Consequently, from a convergence efficiency perspective, the inertia weight should be close to 0.5 to ensure faster optimization.

3.2. Comprehensive performance analysis

The selection of inertia weight w requires balancing optimization quality (best fitness) and convergence efficiency (number of iterations). A very small w value accelerates convergence but may

DOI: 10.54254/2755-2721/2025.KL22614

restrict the particles' exploration ability, leading to premature convergence. Conversely, a very large w value may hinder convergence stability or reduce optimization accuracy.

To determine an optimal inertia weight, we consider the best fitness (w = 0.817) and the fastest convergence (w = 0.5). A compromise approach is adopted by taking the average of these two values:

$$w_{\text{opt}} = \frac{0.5+0.817}{2} = 0.659 \tag{13}$$

From the above analysis, the following conclusions can be drawn:

1. Inertia weight w significantly influences PSO performance; a well-chosen w can improve optimization quality and accelerate convergence.
2. When w = 0.817, the best fitness is minimized, indicating the best solution quality.
3. When w = 0.5, the PSO algorithm converges fastest, but it may suffer from premature convergence.
4. To balance optimization quality and convergence speed, the recommended inertia weight is w = 0.659, ensuring both effective optimization and rapid convergence.

4. The impact of social weight on PSO algorithm performance

4.1. The role of social weight c_1

Social weight c_1 is a crucial parameter in the PSO algorithm, determining the extent to which particles are influenced by the global best solution. A larger c_1 increases the tendency of particles to converge toward the global best, improving overall convergence. However, excessive values of c_1 may weaken the swarm's exploration ability, causing the algorithm to become trapped in local optima. On the other hand, a smaller c_1 enhances individual exploration capability but may lead to slow or unstable convergence. Therefore, an appropriate choice of c_1 is essential for optimizing PSO performance.

In this study, to eliminate the influence of inertia weight on c_1 selection, we fix the inertia weight at w = 0.659 (based on the analysis in Chapter 3) and analyze the effect of c_1 (bounded 1-2.5) on optimization quality and convergence speed in UAV path planning.

4.2. Data fitting analysis

Through experimental data analysis, we establish the relationships between social weight c_1 and the number of iterations and social weight c_1 and best fitness, using quadratic polynomial fitting as Table 2 and Figure 3.

Table 2: Relationship between social weight (c_1), best fitness, and number of iterations

c1	1	1.3	1.5	1.7	2	2.2	2.5
The number of iterations	11.2	10.2	10.5	13.3	10.6	10.8	12.9
Best fitness	162.7246	162.9344	163.1025	162.5227	163.0062	161.9117	161.9286

DOI: 10.54254/2755-2721/2025.KL22614

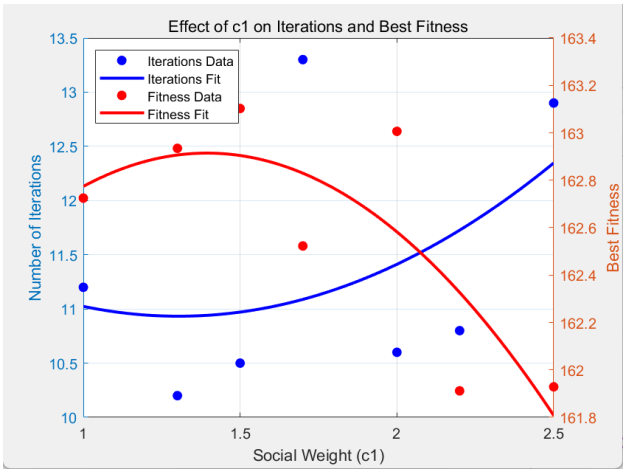

Figure 3: Relationship between social weight (c_1), best fitness, and number of iterations

The resulting equations are:

$$f(c_1) = 0.9872c_1^2 - 2.5735c_1 + 12.6099 \tag{14}$$

$$f(c_1) = -0.9046c_1^2 + 2.5222c_1 + 161.1564 \tag{15}$$

The first equation describes the trend of number of iterations as a function of c_1.

The second equation describes the trend of best fitness as a function of c_1.

From our analysis: The minimum number of iterations occurs at $c_1 = 1.303$, indicating the fastest convergence speed. The maximum best fitness occurs at $c_1 = 1.394$, suggesting that excessive c1 values may deteriorate optimization quality. Additionally, as c_1 increases, best fitness first rises and then falls, while the number of iterations exhibits an overall increasing trend. To balance optimization quality and convergence efficiency, we adopt a trade-off strategy, and the final recommended value of c_1 is:

$$c_{1_{\text{opt}}} = \frac{0.5+0.817}{2} = 1.349 \tag{16}$$

4.3. Conclusions

1. Social weight c_1 influences the global convergence capability of PSO; increasing c_1 enhances global best dependence but may weaken exploration.
2. Best fitness exhibits a nonlinear trend with respect to c_1, reaching a peak near $c_1 = 1.394$. This suggests that at this value, particles become highly dependent on the global best, which may reduce their ability to refine local solutions and affect overall optimization performance. The number of iterations generally increases with c_1, implying that excessive c_1 may reduce convergence efficiency.
3. To balance optimization quality and convergence speed, $c_1 = 1.349$ can achieve optimal PSO performance.

5. The impact of cognitive weight on PSO algorithm performance

5.1. The role of cognitive weight c_2

Cognitive weight c_2 determines the extent to which a particle relies on its personal best solution during the search process. A larger c_2 strengthens local search capability, helping particles converge quickly to individual optima. Conversely, a smaller c_2 increases swarm diversity, enhancing global search

DOI: 10.54254/2755-2721/2025.KL22614

ability. However, excessively high c_2 values may cause particles to concentrate too much on their personal best solutions, reducing overall convergence effectiveness, while excessively low c_2 may lead to inefficient exploration. Therefore, properly adjusting c_2 is critical for optimizing PSO performance.

In this study, we fix the inertia weight at $w = 0.659$ and the social weight at $c_1 = 1.349$ (based on previous chapters) to analyze the impact of c_2 (bounded 1-2.5) on optimization quality and convergence speed in UAV path planning.

5.2. Data fitting analysis

Through experimental data analysis, we establish the relationships between cognitive weight c_2 and the number of iterations and cognitive weight c_2 and best fitness, using quadratic polynomial fitting as Table 3 and Figure 4.

Table 3: Relationship between cognitive weight (c_2), best fitness, and number of iterations

c_2	1	1.3	1.5	1.7	2	2.2	2.5
The number of iterations	12.5	8.7	10.7	11.8	9.7	10.2	10
Best fitness	162.6340	162.8171	162.2872	162.9259	162.0237	162.6596	162.5614

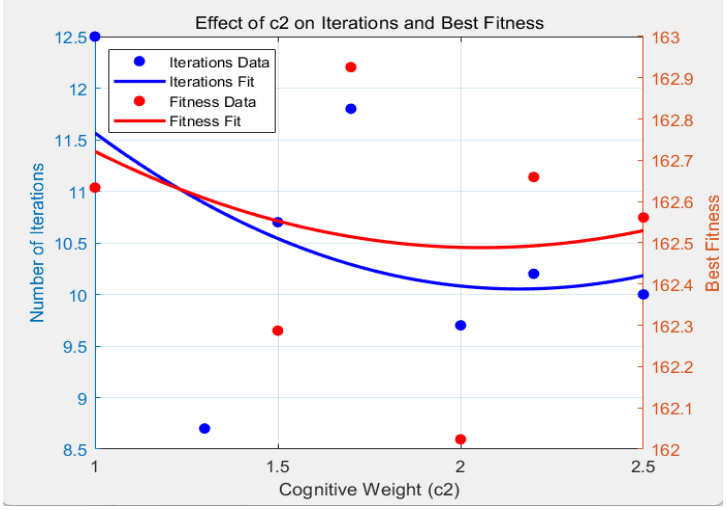

Figure 4: Relationship between cognitive weight (c_2), best fitness, and number of iterations

The resulting equations are:

$$f(c_2) = 1.1248c_2^2 - 4.8585c_2 + 15.2991 \quad (17)$$

$$f(c_2) = 0.2093c_2^2 - 0.8606c_2 + 163.3729 \quad (18)$$

The first equation describes the trend of number of iterations as a function of c_2.

The second equation describes the trend of best fitness as a function of c_2.

From our calculations:

1. The minimum number of iterations occurs at $c_2 = 2.160$, indicating the fastest convergence speed.
2. The minimum best fitness occurs at $c_2 = 2.056$, suggesting that this is the optimal point for achieving the best solution quality.

Furthermore, as c_2 increases, both best fitness and the number of iterations initially decrease and then increase, indicating that excessive c_2 may lead to premature convergence, trapping the algorithm in local optima.

DOI: 10.54254/2755-2721/2025.KL22614

To balance optimization quality and convergence efficiency, we adopt a trade-off strategy, and the final recommended value of c_2 is:

$$c_{2_{\text{opt}}} = \frac{0.5+0.817}{2} = 2.108 \tag{19}$$

5.3. Conclusions and optimization recommendations

1. Cognitive weight c_2 affects the local search ability of PSO; increasing c_2 encourages particles to focus on their personal best solutions, while decreasing c_2 enhances global exploration.
2. Both best fitness and number of iterations exhibit a trend of first decreasing and then increasing with c_2, with best fitness reaching its minimum at $c_2 = 2.056$ and the number of iterations reaching its minimum at $c_2 = 2.160$. This suggests that overly high c2 values may lead to premature convergence.
3. To balance optimization quality and convergence speed, we recommend $c_2 = 2.108$, which ensures better PSO performance.

6. Conclusion

This study delves into the application of particle swarm optimization (PSO) in UAV path planning, with a particular focus on how inertia weight, social weight, cognitive weight, and swarm size influence optimization quality and convergence efficiency. Through a series of simulation experiments and advanced data fitting techniques, mathematical models were developed to elucidate the relationships between these parameters and the algorithm's performance, culminating in the identification of an optimal parameter set. The findings reveal that meticulous parameter tuning significantly bolsters PSO's global search capability, enhances path optimization, and accelerates convergence. Specifically, an optimal inertia weight strikes a balance between global exploration and local refinement, while carefully calibrated social and cognitive weights boost search efficiency. Moreover, an appropriately chosen swarm size ensures an optimal trade-off between search capability and computational cost. The study culminates in the recommendation of the following parameters: inertia weight w = 0.659, social weight c1 = 1.349, cognitive weight $c_2 = 2.108$. These findings not only provide theoretical insights but also offer practical optimization strategies for UAV path planning and other intelligent optimization challenges. Future research could explore adaptive parameter adjustment mechanisms and the integration of multi-objective optimization methods to further enhance PSO's adaptability and computational efficiency in dynamic and complex environments.

References

[1] *Guo, S., Zhao, L., & Wang, H. (2023). Improved PSO with dynamic inertia weight for UAV path planning in urban environments. Sensors, 23(5), 2567.*

[2] *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*

[3] *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*

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

[5] *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*

[6] *Li, M., & Wang, X. (2023). Enhanced PSO-based path planning for UAVs in urban environments with obstacles. Sensors, 23(5), 2678. https://doi.org/10.3390/s23052678*

DOI: 10.54254/2755-2721/2025.KL22614

[7] *Bai, W., Hu, D., Wu, X., Ye, L., Qi, J., & Zhou, Q. (2024). An Improved PSO Algorithm for Hypersonic Vehicle Cruise Trajectory Planning. Lecture Notes in Electrical Engineering, 1186 LNEE, 176–184.*

[8] *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.*

[9] *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.*

[10] *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.*

[11] *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).*

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

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

[14] *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.*

[15] *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.*