Intelligence & Robotics

Research Article

Open Access

Check for updates

Metropolis criterion pigeon-inspired optimization for multi-UAV swarm controller

Jinghua Guan, Hongfei Cheng

School of Computer Science and Technology, Beijing University of Technology, Beijing 100124, China.

Correspondence to: Jinghua Guan, School of Computer Science and Technology, Beijing University of Technology, 100 Pingleyuan, Chaoyang District, Beijing 100124, China. E-mail: gd21110112@emails.bjut.edu.cn

How to cite this article: Guan J, Cheng H. Metropolis criterion pigeon-inspired optimization for multi-UAV swarm controller. *Intell Robot* 2024;4(1):61-73. http://dx.doi.org/10.20517/ir.2024.04

Received: 27 Nov 2023 **First Decision:** 20 Dec 2023 **Revised:** 11 Jan 2024 **Accepted:** 27 Feb 2024 **Published:** 8 Mar 2024

Academic Editors: Haibin Duan, Simon X. Yang **Copy Editor:** Dong-Li Li **Production Editor:** Dong-Li Li

Abstract

This paper presents a new multiple unmanned aerial vehicle swarm controller based on Metropolis criterion. This paper presents the design of a controller, utilizing the improved Metropolis criterion pigeon-inspired optimization (IMCPIO) and proportional-integrational-derivative (PID) algorithms, and conducts comparative experiments. Simulation outcomes demonstrate the enhanced performance of the multi-unmanned aerial vehicle formation controller, which is based on IMCPIO, when compared to the basic pigeon-inspired optimization (PIO) algorithm and the genetic algorithm. The IMCPIO algorithm for the energy difference discrimination makes it a faster convergence and more stable effective optimization. Hence, the controller introduced in this study proves to be both practical and resilient.

Keywords: Pigeon-inspired optimization, Metropolis criterion, unmanned aerial vehicle, formation control, proportional-integrational-derivative

1. INTRODUCTION

There has been a growing trend in the application of multi-unmanned aerial vehicles[1] (UAVs) across a range of military and civil tasks, such as military reconnaissance, surveillance, target identification, search and rescue, and public safety maintenance. Compared to single UAVs, multiple UAVs have significant advantages. For example, during military reconnaissance missions, a single UAV has a limited sensor angle, which cannot cover its task area comprehensively and is vulnerable to enemy interference. Moreover, a single UAV may have insufficient flight range, safety, and attack power, which may compromise its reliability and performance. On the other hand, swarm intelligent systems are efficient and decentralized and can be controlled by a few simple parameters, which enables a single operator to manipulate a large number of UAVs. Furthermore, multiple UAVs can cooperate with each other, adopt dispersed flight paths, reduce the risk of detection and attack, and enhance combat capabilities and task success.

One of the key issues in multi-UAVs systems is to maintain formation among the UAVs during flocking flight. A common strategy for controlling UAV formations is the "Leader-Follower" approach, where one UAV is designated as the Leader, and the rest become Followers. The Leader typically guides the movement of the formation, while the Followers follow the Leader's path and maintain the desired relative position and distance to preserve the formation's shape and collaborative work. In this approach, the trajectory of the 'Leader' is clearly defined, along with the desired relative distance and movement direction between the "Leader" and the "Follower". Although the success of the Leader determines the flight outcome of the entire UAV swarm, this method remains widely used in various fields due to its simplicity, modularity, high fault tolerance, and scalability[2].

In the following section, we will introduce the proportional-integral-derivative (PID) control algorithm, a control strategy extensively employed in automation and control systems due to its simplicity and effectiveness in diverse control situations. The PID algorithm is proficient in executing swift and robust command over the formation of UAVs, ensuring the preservation of their relative positioning, system stability, and resilience to potential faults. However, its linear control and parameter adjustment methods often fall into local optima. The metaheuristic algorithm provides a solution to the above problems. Duan and Qiao proposed a novel optimization algorithm, pigeon-inspired optimization (PIO), which draws inspiration from the behavior of pigeons[3]. This algorithm has found applications in solving optimization problems, including UAV path planning[4] and image recognition[5]. It comprises two key components: the map and compass operator and the landmark operator[6]. However, the basic PIO algorithm also tends to get trapped in local optima and has a slow convergence speed, which is not suitable for multi-UAVs formation scenarios.

Therefore, we propose an improved PIO algorithm, called improved Metropolis criterion PIO (IMCPIO), inspired by the simulated annealing (SA) algorithm[7] and the Iterative Modified PSO (IMPSO) algorithm. The IMPIO algorithm has the following advantages over the basic PIO algorithm: (1) It allows inferior solutions to be accepted with a certain probability, which enables the algorithm to escape local optima and enhances its robustness; (2) It introduces a temperature parameter T, which decreases gradually. This implies that during the initial phases of the algorithm, a greater likelihood of accepting suboptimal solutions aids in avoiding local peaks. As the temperature decreases, the algorithm is more likely to accept solutions that are slightly worse than the current solution, which helps the algorithm converge to the global optimum; (3) It adds a speed halving strategy, which controls the range of particle movement, improves search accuracy, and makes particles more stable near the global optimum. This reduces unnecessary jumps and is conducive to fine-tuning the solution; (4) It adopts a correction strategy from the IMPSO algorithm, proposed by Yang *et al*., which fixes the defect that particles easily fall into local optima and hover near the optimal position[8].

The subsequent sections of this paper are structured in the following manner. The first part of Section 2 describes the mathematical model of a multi-UAV formation controller. The second part briefly reviews the basic PIO algorithm, while the third part introduces the IMCPIO algorithm, which is an improved version of the PIO algorithm. Comparative simulations are performed in Section 3. Section 4 summarizes the paper and discusses future work.

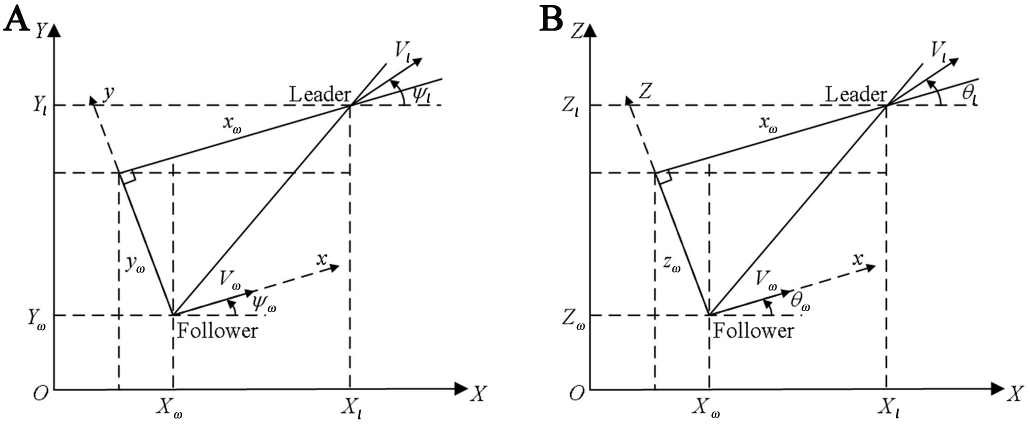

Figure 1. Inertial coordinate system. (A) Inertial coordinate systems for x and y axes; (B) Inertial coordinate system for x and z axes.

2. METHODS

2.1. Swarm with multiple UAVs: mathematical model and built-in controller

In this paper, we utilize a "leader-follower" model with two UAVs[9] and exemplify it with an inertial coordinate system[10]. As illustrated in Figure 1[11,12], the follower is designated as the origin point for the establishment of a reference coordinate system. As depicted in Figure 1A and B, the O-XYZ represents the inertial coordinate system. First, defineX_l, Y_l and Z_l as the leader's position within the inertial coordinate system,V_l as the velocity, ψ_l as the heading angle, and θ_l as the pitching angle. Similarly, define X_ω, Y_ω, and Z_ω as the coordinates of the follower in the inertial coordinate system, V_ω as the velocity, ψ_ω as the heading angle, and θ_ω as the pitching angle, along with X_ω, Y_ω, and Z_ω as the distances between the leader and the follower[13].

The autopilot governs the movement of both the leader and follower UAVs. This control is based on a mathematical model:

$$\begin{aligned}
V_l' &= \frac{1}{\tau_{V_l}}\left(V_{lc} - V_l\right),\\
V_\omega' &= \frac{1}{\tau_{V_\omega}}\left(V_{\omega c} - V_\omega\right),\\
\psi_l' &= \frac{1}{\tau_{\psi_l}}\left(\psi_{lc} - \psi_l\right),\\
\psi_\omega' &= \frac{1}{\tau_{\psi_\omega}}\left(\psi_{\omega c} - \psi_\omega\right),\\
\theta_l' &= \frac{1}{\tau_{\theta_l}}\left(\theta_{lc} - \theta_l\right),\\
\theta_\omega' &= \frac{1}{\tau_{\theta_\omega}}\left(\theta_{\omega c} - \theta_\omega\right).
\end{aligned} \tag{1}$$

where $\tau_{V_l}, \tau_{V_\omega}, \tau_{\psi_l}, \tau_{\psi_\omega}, \tau_{\theta_l}, \tau_{\theta_\omega}$ are identified as the temporal constants for velocity, heading angle, and pitching angle. The formation controller's goal, which operates on the IMPIO algorithm, is to uphold a specific separation between the leading and following UAVs. This is achieved by inputting the follower's control instruction $V_{\omega c}$, $\psi_{\omega c}$ and $\theta_{\omega c}$.

$$\begin{aligned}
X_l' &= V_l \cos\psi_l \cos\theta_l,\\
Y_l' &= V_l \sin\psi_l \cos\theta_l,\\
Z_l' &= V_l \sin\theta_l,\\
X_\omega{}' &= V_\omega \cos\psi_\omega \cos\theta_\omega,\\
Y_\omega{}' &= V_\omega \sin\psi_\omega \cos\theta_\omega,\\
Z_\omega{}' &= V_\omega \sin\theta_\omega.
\end{aligned} \tag{2}$$

Based on the relative position of the follower and leader in Figure 1A and B, we can formulate the position of the leader UAV as follows:

$$\begin{aligned} X_l &= X_\omega + x_\omega \cos\psi_\omega \cos\theta_\omega - y_\omega \sin\psi_\omega + z_\omega \cos\psi_\omega \sin\theta_\omega, \\ Y_l &= Y_\omega + x_\omega \sin\psi_\omega \cos\theta_\omega + y_\omega \cos\psi_\omega + z_\omega \sin\psi_\omega \sin\theta_\omega, \\ Z_l &= Z_\omega + x_\omega \sin\theta_\omega + z_\omega \cos\theta_\omega. \end{aligned} \tag{3}$$

In the reference coordinate system, where x_ω, y_ω,and z_ω are, the representation can be depicted as follows:

$$\begin{bmatrix} x_\omega \\ y_\omega \\ z_\omega \end{bmatrix} = A^{-1} \begin{bmatrix} X_l - X_\omega \\ Y_l - Y_\omega \\ Z_l - Z_\omega \end{bmatrix} \tag{4}$$

where the matrix is:

$$A = \begin{bmatrix} \cos\psi_\omega \cos\theta_\omega & -\sin\psi_\omega & \cos\psi_\omega \sin\theta_\omega \\ \sin\psi_\omega \cos\theta_\omega & \cos\psi_\omega & \sin\psi_\omega \sin\theta_\omega \\ \sin\theta_\omega & 0 & \cos\theta_\omega \end{bmatrix} \tag{5}$$

By considering the position of the leader UAV and the desired inter-drone distance, we can compute the expected position of the follower UAV in the inertial coordinate system. The relative expected position of the follower is denoted by $\hat{X}_\omega$, $\hat{Y}_\omega$ and $\hat{Z}_\omega$. The error between the current position of the follower and the desired position is defined as:

$$e = \begin{bmatrix} X_l - X_\omega \\ Y_l - Y_\omega \\ Z_l - Z_\omega \end{bmatrix} = \begin{bmatrix} X_l - \hat{X}_\omega \\ Y_l - \hat{Y}_\omega \\ Z_l - \hat{Z}_\omega \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} J_1 \\ J_2 \\ J_3 \end{bmatrix} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} X_l - X_\omega \\ Y_l - Y_\omega \\ Z_l - Z_\omega \end{bmatrix} = \begin{bmatrix} X_l - \hat{X}_\omega \\ Y_l - \hat{Y}_\omega \\ Z_l - \hat{Z}_\omega \end{bmatrix} \tag{7}$$

Assume that each following drone is equipped with three PID controllers to control its speed V_ω, heading angle ψ_ω and pitch angle θ_ω. Thes econtrollers can generate the control inputs $V_{\omega c}, \psi_{\omega c}$, and $\theta_{\omega c}$ of the following drone based on the errors e_x, e_y and e_z. The formula is as follows:

$$\begin{aligned} V_{\omega c} &= K_{px} e_x + K_{ix} \int e_x dt + K_{dx} \frac{de_x}{dt}, \\ \psi_{\omega c} &= K_{py} e_y + K_{iy} \int e_y dt + K_{dy} \frac{de_y}{dt}, \\ \theta_{\omega c} &= K_{pz} e_z + K_{iz} \int e_z dt + K_{dz} \frac{de_z}{dt}. \end{aligned} \tag{8}$$

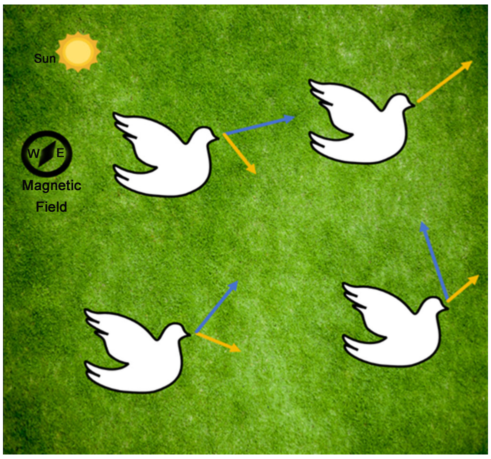

Figure 2. Map and compass operator in PIO. PIO: Pigeon-inspired optimization.

2.2. PIO principles

Drawing inspiration from the distinctive navigational behavior of pigeon flocks during their homing process, we put forth a bionic population intelligence algorithm, termed as the PIO algorithm[14]. This algorithm employs distinct operators at different stages, specifically the map and compass operator and the landmark operator[15]. The map and compass operator encapsulates the influence of the magnetic field on the pigeon's flight direction, while the landmark operator represents the impact of landmarks on its navigational trajectory.

Suppose a flock of pigeons numbered N searches for a target in a d-dimensional space. When is considered, the map and compass operator identifies each unique pigeon as "i", which stands for the maximum number of iterations that the map and compass operator can perform. Its spatial coordinates and speed are symbolized as follows:

$$\begin{aligned} X_i &= [x_{i1}, x_{i2}, ..., x_{id}], \\ V_i &= [v_{i1}, v_{i2}, ..., v_{id}]. \end{aligned} \tag{9}$$

when the number of iterations is N_c; each pigeon updates its position $X_i^{N_c}$ and $V_i^{N_c}$ velocity according to the following formulas:

$$\begin{aligned} V_i^{N_c} &= V_i^{N_c-1} e^{-R \times N_c} + rand(X_g - X_i^{N_c-1}), \\ X_i^{N_c} &= X_i^{N_c-1} + V_i^{N_c}. \end{aligned} \tag{10}$$

In this context, R represents the map and compass operator, which has a value range from 0 1. Similarly, $rand()$ is a function that generates a random number within the same range, 0 1. X_g is the global optimal position obtained by comparing the positions of all pigeons after $N_c - 1$ iterative loop. In accordance with Equation 9, each pigeon has the ability to modify its flight direction. This is represented by the blue arrow, while the yellow one signifies the original direction. The subsequent direction of the pigeon is determined by the vector sum of the blue and yellow directions. Figure 2 gives an example of the action of pigeons in the map and compass operator phase.

As shown in Figure 3, when the above loop reaches the maximum number of iterations of the phase, i.e., $N_{c_{max}}^1 < N_c < N_{c_{max}}^2$ the map and compass operator stops working and enters the land-mark operator phase,

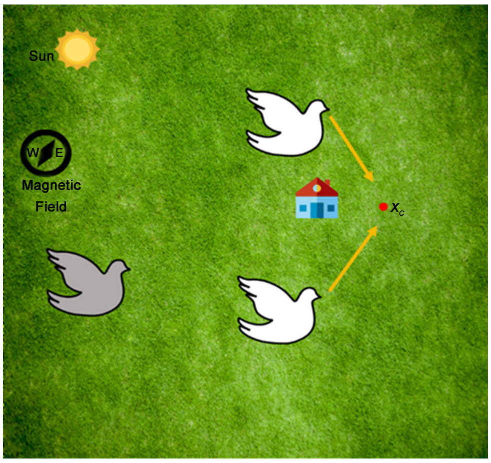

Figure 3. Landmark operator in PIO. PIO: Pigeon-inspired optimization.

where $N^2_{c_{max}}$ is the maximum number of iterations of the landmark operator. As depicted in the figure, the landmark operator reduces the number of pigeons (gray) by half after each iteration. Pigeons distant from the destination lose their path-distinguishing ability due to unfamiliarity with the landmarks and are consequently discarded. Conversely, those near the destination (white) quickly orient themselves towards their target.

In this phase, the center position of the flock, constituted by the remaining pigeons, serves as a landmark. This landmark provides a reference direction for the flight path of the remaining pigeons. The position update equation for pigeon i is as follows:

$$\begin{aligned} &N = floor\left(\frac{N}{2}\right), \\ &X_c^{N_c-1} = \frac{\sum_{i=1}^{N} X_i^{N_c-1} F\left(X_i^{N_c-1}\right)}{\sum_{i=1}^{N} F\left(X_i^{N_c-1}\right)}, \\ &X_i^{N_c} = X_i^{N_c-1} + rand\left(X_c^{N_c-1} - X_i^{N_c-1}\right). \end{aligned} \tag{11}$$

where $floor\left(\frac{N}{2}\right)$ is the ceiling function.

$$f(x) = \begin{cases} fitness\left(X_i^{N_c-1}\right), & \text{for maximization problems,} \\ \frac{1}{fitness\left(X_i^{N_c-1}\right)+\varepsilon}, & \text{for minimization problems.} \end{cases} \tag{12}$$

where $fitness\left(X_i^{N_c-1}\right)$ is the cost function of pigeon i at the sub-iteration.

2.3. PIO principles

In this section, we introduce a novel approach, termed IMCPIO, for managing the PIO algorithm. This method is grounded in the work of Sun and Duan[16]. Although the PIO base algorithm has advantages such as higher robustness, it still faces problems of being prone to falling into local optimal solutions and slower convergence and it is not applicable to UAV formation scenarios. Inspired by the SA algorithm, the improved Metropolis criterion (IMC) prevents the particles from falling into local optimal solutions. The Metropolis criterion makes certain choices for handling iterative updates. If the energy of the next iteration is low, it is updated directly to the next position. When the energy of the next iteration is higher, a certain probability to iterate also exists. The Metropolis criterion compares the energy of the current state with that of the next step and calculates the probability of iteration:

$$f(x)\begin{cases}1, \Delta E < 0,\\ \exp(-\Delta E/T), \Delta E > 0.\end{cases} \tag{13}$$

where $\Delta E = fitness\left(X_i^{N_c}\right) - fitness\left(X_i^{N_c-1}\right)$ is the difference between the fitness value of the current iteration and the fitness value of the previous iteration. T is the system temperature, which decreases as the number of iterations increases.

The base Metropolis criterion has a reduced convergence speed due to the particle staying at the last position. Therefore, an IMC proposed by Yang *et al.* is introduced and applied to IMCPIO[8]. For the method of updating the position X_i^{Nc} and velocity V_i^{Nc} of each pigeon in the first iteration, the improvement steps are shown in Figure 4. The specific implementation of the IMCPIO algorithm that introduces the IMC is as follows:

Step 1. Initialize the airspace information and the dangerous areas information.

Step 2. Initialize IMCPIO algorithm parameters, including space dimension d, population size N_c, map and compass factor R, the number of $N_{c_{max}}^1$ and $N_{c_{max}}^2$ for two operators, *etc.*

Step 3. Allocate a random position and velocity to each pigeon. Subsequently, the position X_g, which represents the best global value, is determined by comparing the fitness of each pigeon.

Step 4. Execute the map and compass operator. Refresh the velocity and trajectory of each pigeon utilizing Equation (9). The updated positions of individuals within the boundary are filtered using the improved Metropolis criterion. If the energy of the next iteration is low, it is updated directly to the next position. When the energy of the next iteration is higher, there is also a certain probability to iterate. At the end of the position update operation for each individual, evaluate the local optimal positions, compare the fitness of each pigeon, and determine the updated X_g.

Step 5. If the iteration count exceeds $N_{c_{max}}^1$, halt the map and compass operator and initiate the landmark operator. If not, proceed to Step 4.

Step 6. All pigeons are sorted based on their fitness value. The half with higher fitness values will follow those with lower fitness. Using Equation (10), compute $X_C^{N_c}$ and update the position $X_i^{N_c}$. If the iteration count exceeds $N_{c_{max}}^2$, the landmark operator is halted. If not, return to Step 6.

3. RESULTS

To evaluate the performance of the proposed IMPIO algorithm, we used a set of benchmark functions that have different characteristics and compared it with the basic PIO method and the genetic algorithm (GA) method. The benchmark functions are: Sphere (*f1*), which is a simple unimodal function that measures the basic performance of the algorithm, such as convergence speed and accuracy; Rosenbrock (*f2*), which is a nonlinear multimodal function that measures the algorithm's ability to optimize in high-dimensional spaces and escape from local optima; Ackley (*f3*), which is a multimodal function with one global optimum and

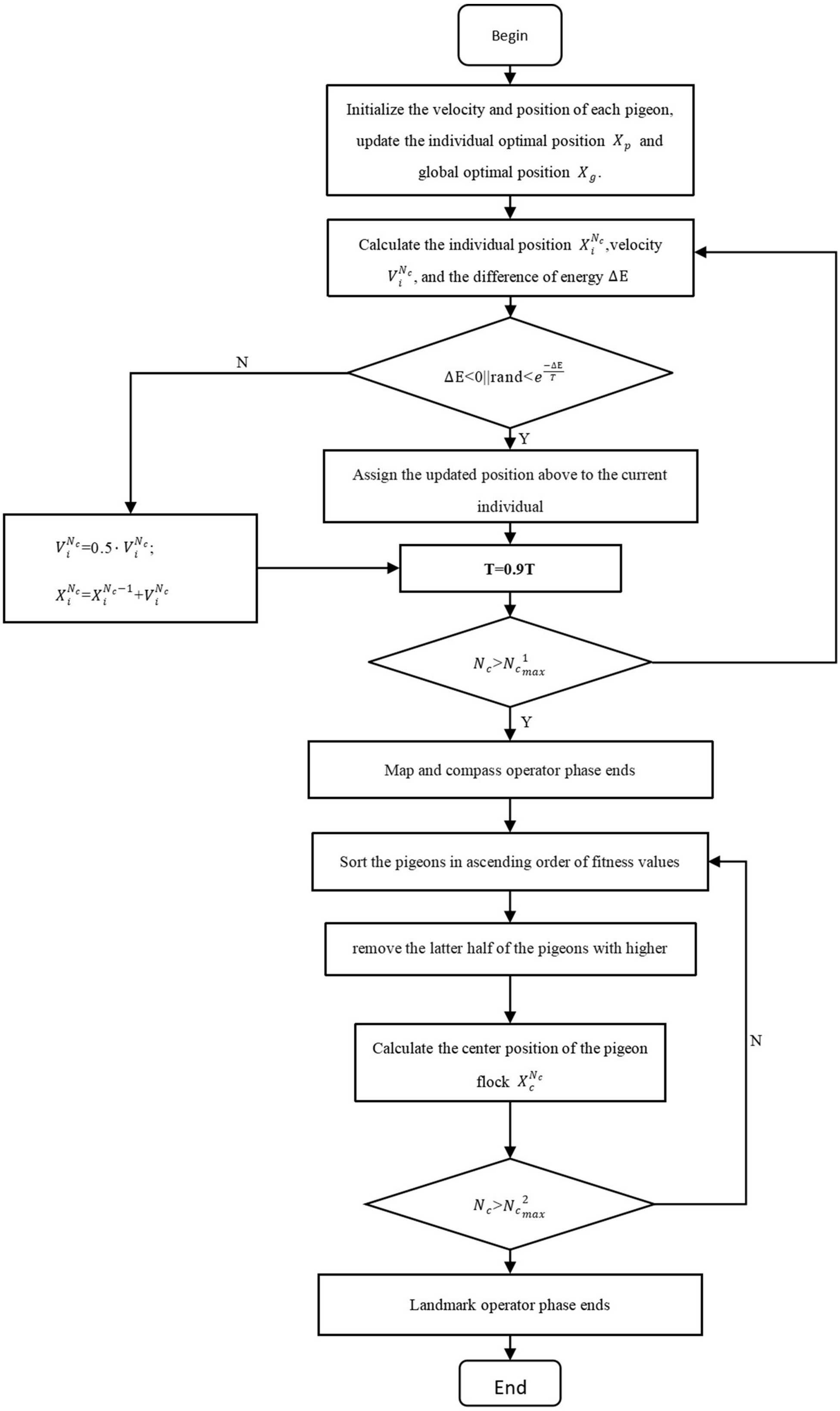
Begin
Initialize the velocity and position of each pigeon, update the individual optimal position X_p and global optimal position X_g.
Calculate the individual position $X_i^{N_c}$,velocity $V_i^{N_c}$, and the difference of energy ΔE
$\Delta E<0||rand<e^{\frac{-\Delta E}{T}}$
N
Y
Assign the updated position above to the current individual
$V_i^{N_c}=0.5\cdot V_i^{N_c}$;
$X_i^{N_c}=X_i^{N_c-1}+V_i^{N_c}$
T=0.9T
$N_c>N_{c_{max}}^1$
N
Y
Map and compass operator phase ends
Sort the pigeons in ascending order of fitness values
remove the latter half of the pigeons with higher
Calculate the center position of the pigeon flock $X_c^{N_c}$
N
$N_c>N_{c_{max}}^2$
Landmark operator phase ends
End

Figure 4. Flowchart of IMCPIO. IMCPIO: Improved Metropolis criterion pigeon-inspired optimization.

Table 1. Benchmark functions

Function	Common name	Expression
f_1	Sphere	$f_1(x) = \sum_{i=1}^{d} x_i^2$
f_2	Rosenbrock	$f_2(x) = \sum_{i=1}^{d} 100\left(\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2\right)$
f_3	Ackley	$f_3(x) = -20e^{-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}} - e^{\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)} + 20 + e$
f_4	Schwefel	$f_4(x) = \sum_{i=1}^{d}\left(\sum_{j=1}^{i} x_j\right)^2$
f_5	Rastrigin	$f_5(x) = \sum_{i=1}^{d}\left(x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right)$

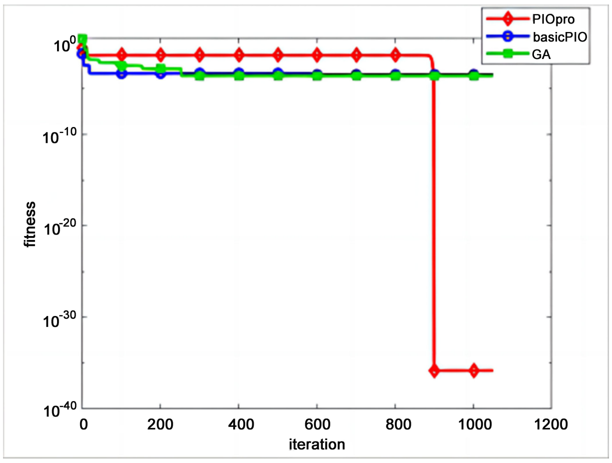

Figure 5. Sphere function comparison curves. PIO: Pigeon-inspired optimization; GA: genetic algorithm.

many local optima, which measures the algorithm's ability to optimize in complex environments and stability; Griewank (*f4*), which is a multimodal function with many local optima, which measures the algorithm's ability to optimize in high-frequency oscillations and robustness; Rastrigin (*f5*), which is a multimodal function with many local optima, which measures the algorithm's global search and anti-interference abilities.

The functional characteristics of the above five functions are shown in Table 1.

In order to better demonstrate the advantages of IMCPIO, the IMCPIO algorithm is compared with the basic PIO and GA algorithms, and the result curves are as follows:

Figures 5-9 demonstrate the search capabilities of the three algorithms for the five test functions in Table 1. In the case of both the IMCPIO and base PIO algorithms, the map compass operator stage was configured to perform 900 iterations, while the surface operator stage was set to execute 150 iterations. The results indicate that the IMCPIO algorithm exhibits superior performance when dealing with multidimensional optimization problems. Compared with the basic PIO and GA algorithms, the IMCPIO algorithm has significantly enhanced convergence speed, improved ability to escape local optima, and a significantly reduced final best fitness value.

The introduction of the Metropolis criterion in the IMCPIO algorithm effectively handles local optima in optimization problems, enhancing its ability to escape local optima. The two-stage search strategy employed by the IMCPIO algorithm allows for a broad search in the initial stage to locate the approximate position of the global optimum and then a fine search in the later stage to precisely locate the global optimum. The implementation of this strategy results in a substantial enhancement in the convergence speed of the IMCPIO algorithm. The multi-agent characteristic of the IMCPIO algorithm, where each pigeon can independently conduct a search, enables it to effectively handle high-dimensional search spaces. Therefore, its optimization capability surpasses that of the basic PIO and GA algorithms. This aligns with the simulation results, further validating the superiority of the IMCPIO algorithm.

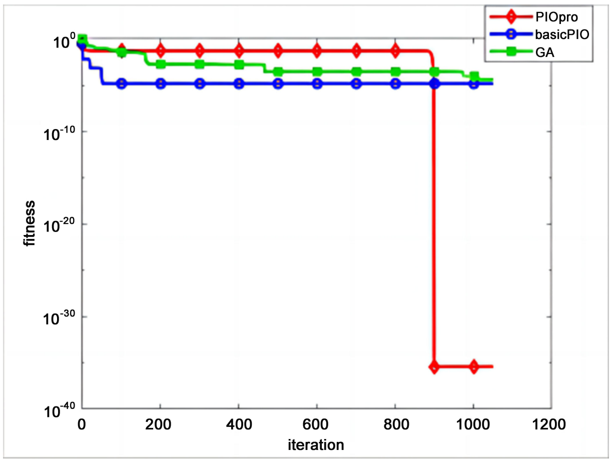

Figure 6. Rosenbrock function comparison curves. PIO: Pigeon-inspired optimization; GA: genetic algorithm.

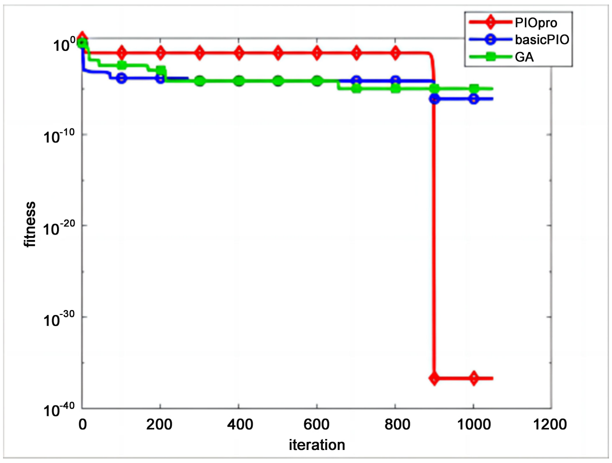

Figure 7. Ackley function comparison curves. PIO: Pigeon-inspired optimization; GA: genetic algorithm.

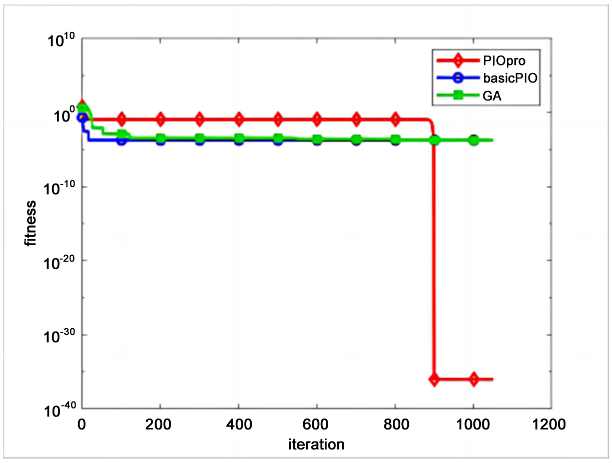

Figure 8. Schwefel function comparison curves. PIO: Pigeon-inspired optimization; GA: genetic algorithm.

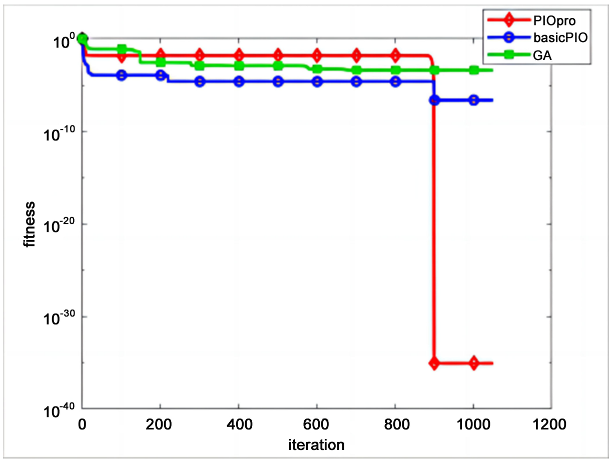

Figure 9. Rastrigin function comparison curves. PIO: Pigeon-inspired optimization; GA: genetic algorithm.

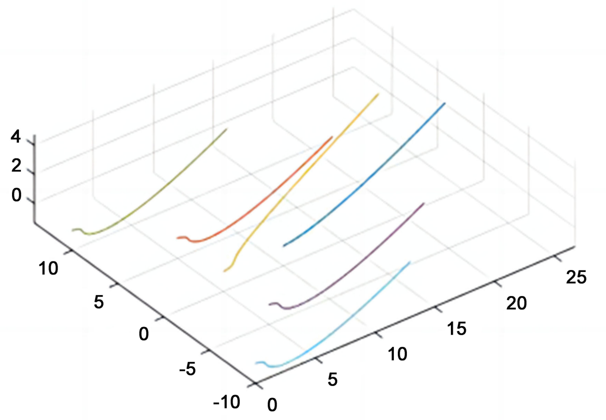

Figure 10. Multi-UAVs three-dimensional trajectories. UAVs: Unmanned aerial vehicles.

Figure 10 depicts the detailed results of six UAVs in formation light in a three-dimensional environment.

4. DISCUSSION

This paper proposes an IMCPIO algorithm that introduces the Metropolis criterion based on the basic PIO algorithm and combines it with the PID algorithm to optimize controller parameters. Simulation results demonstrate that the IMCPIO algorithm significantly improves the convergence speed and the ability to escape local optima compared to the basic PIO algorithm and the GA, ultimately enhancing the optimization effect.

Currently, this algorithm only introduces the IMC in the map and compass operator stage of the base PIO algorithm. Looking forward, there is potential for introducing more advanced optimization strategies into IMCPIO to further refine both the map and compass operator stage and the landmark operator stage. This would further enhance the ability of the IMCPIO algorithm to escape local optima and its convergence. Moreover, the integration of the IMCPIO algorithm with the PID algorithm opens up new avenues for optimization. The balance between global and local search in the IMCPIO algorithm can be used to adaptively adjust the parameters of the PID controller, enhancing its performance. This combination could also improve the handling of non-linear systems and uncertainties, which are common in practical applications.

In the future, this algorithm will be further refined to enhance its optimization capability, such as escaping from local optimum solutions, so that it can better serve the optimization of controller parameters. This presents an exciting direction for future work. Subsequent research is needed to validate these prospects, but the integration of these two algorithms could potentially provide a powerful tool for tackling complex optimization problems.

DECLARATIONS

Authors' contributions

Significantly contributed to the conceptualization of the study and the methodology proposed and performed the validation, analysis, investigation, resource acquisition, and writing: Guan J
Performed article review and editing, project supervision and management: Cheng H

Availability of data and materials

Not applicable.

Financial support and sponsorship

None.

Conflicts of interest

Both authors declared that there are no conflicts of interest

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Tong B, Wei C, Shi Y. Fractional order darwinian pigeon-inspired optimization for multi-UAV swarm controller. *Guid Navig Control* 2022;2:2250010. DOI
2. Consolini L, Morbidi F, Prattichizzo D, Tosques M. Leader-follower formation control of nonholonomic mobile robots with input constraints. *Automatica* 2008;44;1343-49. DOI
3. Duan H, Qiao P. Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. *Int J Intell Comput Cybernet* 2014;7;24-37. DOI
4. Zhang B, Duan H. Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment. *IEEE/ACM Trans Comput Biol Bioinformat* 2017;14:97-107. DOI
5. Duan H, Wang X. Echo state networks with orthogonal pigeon-inspired optimization for image restoration. *IEEE Trans Neural Netw Learn Syst* 2016;27;2413-25. DOI
6. Duan H, Qiu H. Advancements in pigeon-inspired optimization and its variants. *Sci Chin Informat Sci* 2019;62:70201. DOI
7. Hu Q, Zhang MS. A collaborative optimization for floorplanning and pin assignment of 3D ICs based on GA-SA algorithm. In: 2020 IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI); 2020 Jul 28 - Aug 28; Reno, NV, USA. IEEE; 2020. pp.434-8. DOI
8. Yang C, Chen R, Wang W, Li Y, Shen X, Xiang C. Cyber-physical optimization-based fuzzy control strategy for plug-in hybrid electric buses using iterative modified particle swarm optimization. *IEEE Trans Intell Veh* 2023;8:3285-98. DOI
9. Rezaee H, Abdollahi F, Menhaj MB. Model-free fuzzy leader-follower formation control of fixed wing UAVs. In: 2013 13th Iranian Conference on Fuzzy Systems (IFSC); 2013 Aug 27-29; Qazvin, Iran. IEEE; 2013. pp.1-5. DOI
10. Tong B, Chen L, Duan H. A path planning method for UAVs based on multi-objective pigeon-inspired optimisation and differential evolution. *Int J Bio-Inspired Comput* 2021;17;105-12. DOI
11. Zong L, Xie F, Qin S. Intelligent optimizing control of formation flight for UAVs based on MAS. *Acta Aeronaut Astronaut Sin* 2008;29:1326-33. Available from: https://hkxb.buaa.edu.cn/EN/abstract/abstract9316.shtml. [Last accessed on 4 March 2024]

12. Tousi SMA, Mostafanasab A, Teshnehlab M. Design of self tuning PID controller based on competitional PSO. In: 2020 4th Conference on Swarm Intelligence and Evolutionary Computation (CSIEC); 2020 Sep 2-4; Mashhad, Iran. IEEE; 2020. pp. 22-6. DOI
13. Zhang X, Duan H, Yang C. Pigeon-Inspired optimization approach to multiple UAVs formation reconfiguration controller design. In: Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference; 2014 Aug 8-10; Yantai, China. IEEE; 2014. pp. 2707-12. DOI
14. Xu Z. Study on detection online of PIO for shipboard UAV based on neural network. In: 2022 China Automation Congress (CAC); 2022 Nov 25-27; Xiamen, China. IEEE; 2022. pp. 3641-5. DOI
15. Chen Z, Wei C, Zheng Y. A cooperative approach to multi-UAVs search for mobile targets based on pigeon-inspired optimization. In: 2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC); 2018 Aug 10-12; Xiamen, China. IEEE; 2018. pp. 1-8. DOI
16. Sun H, Duan H. PID controller design based on Prey-Predator Pigeon-Inspired Optimization algorithm. In: 2014 IEEE International Conference on Mechatronics and Automation. 2014 Aug 3-6; Tianjin, China. IEEE; 2014. pp. 1416-21. DOI