The Journal of Supercomputing (2023) 79:15997–16016
https://doi.org/10.1007/s11227-023-05246-8

The HSGWO-MPIO algorithm based on improved search capability

Xinrong Zhou[1] · Fang Wang[1] · Chao Zhou[2] · Rui Shan[1]

Accepted: 2 April 2023 / Published online: 24 April 2023

Abstract
This paper presents a hybrid simplified grey wolf optimization algorithm combining with an modified pigeon inspired optimization algorithm. Firstly, a method of population initialization in the restricted area is designed to improve the average fitness of individuals. Then, a decay factor is designed to improve the search accuracy. At the same time, a new search position update formula is designed to balance the global search and local search ability, so as to improve the convergence speed of the algorithm. Furthermore, in order to verify the performance, the convergence, complexity and accuracy of the algorithm are analyzed by using linear difference equations. Finally, the comparison between different meta-heuristic optimization algorithms, the influence of parameters and their application in the path planning of unmanned aerial vehicle (UAV) are tested. Simulation results show that the proposed algorithm has stronger optimization ability and better robustness than those of compared algorithms.

Keywords Grey wolf optimization (GWO) algorithm · Pigeon inspired optimization (PIO) algorithm · Hybrid optimization algorithm · Decay factor · Convergence rate

1 Introduction

As we known that intelligent optimization algorithms have been widely used in various fields for their efficient performance advantages. They play a crucial role in the refueling optimization scheduling of multiple UAVs [1], express delivery efficiency [2], intelligent robot path optimization [3] and intelligent navigation [4]. With the

The contribution of lead author Xinrong Zhou and corresponding author Fang Wang in this article is equal.

✉ Fang Wang
wangfang@ysu.edu.cn

1 School of Science, Yanshan University, Qinhuangdao 066004, Hebei, China

2 Ocean College, Hebei Agricultural University, Qinhuangdao 066003, Hebei, China

development of big data and artificial intelligence, many complex and challenging problems need to be solved.

For Non-deterministic Polynomial (NP) hard problems, there are two kinds of research methods commonly used to solve: One is heuristic method, such as artificial potential field (APF) method [5], simulated annealing (SA) algorithm [6]. Choi et al. [7] proposed a hybrid algorithm based on the genetic algorithm (GA) and the evolution strategy (ES) for the electromagnetic optimization problem. In order to make the algorithm more suitable for disaster scenarios, an adaptive selection mutation constrained differential evolution algorithm [8] was proposed. It selects individuals according to individual fitness and constraints. Alkhateeb et al. [9] proposed discrete azalea search and simulated annealing (CSA) algorithm to solve continuous optimization problems. The heuristic optimization algorithm has the advantages of fast planning speed, good parallelism, and easy operation. Another is swarm intelligence optimization algorithm, such as particle swarm optimization (PSO) algorithm [10], ant colony optimization (ACO) algorithm [11]. A modified PSO algorithm has been proposed by Lee et al. [12]. The ACO algorithm [13, 14] has been introduced to adaptively optimize the clustering number of position data and solve the over-fitting problem of the single k-means algorithm. An improved whale optimization algorithm (IWOA) based on logic mapping [15] was designed to perform parameters identification. Unlike many traditional optimization algorithms, it relies on the instinct of each organism to optimize the survival state through unconscious evolution and optimization behavior to adapt to different task environments. The robustness and convergence speed of the algorithm are always the focus of the optimization algorithm. More and more research is devoted to achieving a balance between the convergence and search ability of the algorithm [16–18]. The swarm intelligence optimization algorithm is based on the research of social insect group behavior. The macroscopic intelligent behavior characteristics of social organisms that achieve the goal of optimization through cooperation, information exchange and cooperation between groups and simple and seemingly individual interaction have received more and more attention.

Inspired by grey wolves, grey wolf optimization (GWO) algorithm was proposed [19]. It mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. However, it is prone to stagnate in local solutions, and the convergence speed may be slower. A four steps decision wolf optimization algorithm and a GWO algorithm combined with PSO algorithm [20, 21] were proposed to overcome the shortcomings. To solve the problem that the traditional GWO algorithm easily falls into the local minima when solving complex optimization problems, an improved GWO algorithm based on the two-stage search of hybrid covariance matrix adaptation-evolution strategy (CMA-ES) was proposed by Zhao et al. [22]. In order to improve the convergence rate in the minimum period, Dev et al. [23] proposed a hybrid rider optimization algorithm (ROA) and GWO algorithm. It is proved that the optimal location of solution space can be searched quickly and effectively by using species evolution mechanism and parallel optimization method [24]. The GWO algorithm has been applied in many engineering applications, such as economic load dispatch problems [25], multi-tracking target [26], wide-area power system stabilizer design [27]. In the proposed algorithm [20–23], the exploration and exploitation abilities were combined efficiently. However, the convergence accuracy of the algorithm is not further analyzed. In this paper, we use the linear difference equation method to comprehensively analyze the convergence speed and convergence accuracy of the proposed algorithm.

Pigeon inspired optimization (PIO) algorithm was originally developed by Duan and Qiao [28]. It is a population optimization algorithm simulating the homing behavior of pigeons. Compared with other optimization algorithms, PIO algorithm has the advantages of simple principle, less adjustment parameters, easy implementation and strong robustness. Duan et al. [29] simplified the multi-objective PIO algorithm by considering the factors that affect the flight of pigeons. Mutation mechanism was introduced to enhance the exploration ability in the process of evolution. However, the computational complexity is not explained. After that, in order to better solve the practical application problem, Duan et al. [30] proposed a dynamic discrete pigeon heuristic optimization algorithm to deal with the cooperative search and attack mission planning of UAVs. This method focuses on the practical application scenario and dynamically demonstrates the coordinated flight process of UAVs. Wu et al. [31] introduced Gaussian mutation into the basic PIO algorithm to maintain the diversity of the group and avoid premature convergence. Then, the optimization of flight control system is considered. The existing algorithms focus on improving the PIO algorithm itself and apply it in the aerospace field. However, to our best knowledge, there is rarely result about combining PIO algorithm with other algorithms and realizing their complementary advantages. In this paper, we designed the HSGWO-MPIO algorithm to reflect the advantages of the algorithm combination.

This paper not only overcomes the problems of large search range and fast maturity of optimization algorithm, but also solves the problems of weak group search ability. The search accuracy and computational complexity are also considered. Hybrid optimization algorithm is the combination of simplified GWO algorithm and improved PIO algorithm. It transforms the original problem into the problem of verifying the feasibility of the algorithm under the new algorithm. The main contributions of this paper are summarized as follows:

(1) A new decay factor is designed to simplified the GWO algorithm where the convergence speed of the simplified GWO algorithm is improved and the global optimization accuracy is maintained.
(2) Considering the computational complexity, the location updating formula of PIO algorithm is improved, which enriches the diversity of the population and increases the search range of the population.
(3) The convergence of the linear difference equation is analyzed theoretically, and the spatial complexity of the algorithm is analyzed to obtain the optimal global solution. The rationality, effectiveness and feasibility of the algorithm are verified by the comparison of algorithm parameters, algorithm convergence and UAV flight path.

The rest of this paper is arranged as follows. Section 2 presents a hybrid optimization algorithm and analyzes the convergence, complexity and accuracy of the algorithm. Section 3 shows the effectiveness of the method by compared simulations. Finally, Sect. 4 concludes the paper.

2 The hybrid SGWO-MPIO algorithm

2.1 Basic ideas of the proposed algorithm

To solve complex optimization problems, all the meta-heuristic algorithms like GWO algorithm and PIO algorithm have been designed. In GWO algorithm, the positions of individuals are updated according to the α wolf, β wolf and δ wolf. Meanwhile, the exploration ability and the exploitation ability mainly depend on the parameter A. When the random values of A are in $[-1, 1]$, the wolves move with smaller distance, which means a process of local search.

In PIO algorithm, the update of the organisms' position is not only related to the current optimal organism but also affected by random numbers in the population. In this way, all possible solutions can be searched during the optimization process. Consequently, the optimization pattern of the GWO algorithm gives it a great exploration ability. In contrast, the PIO algorithm has an aptitude for local exploitation capability.

Therefore, to solve the complex problems, we have simplified the GWO algorithm to retain the exploration ability and accelerate the convergence speed. Then, the PIO algorithm is modified to enhance the exploitation capability. Finally, to combine the advantages of simplified GWO (SGWO) and modified PIO (MPIO) algorithm, we present a novel hybrid algorithm called the HSGWO-MPIO algorithm.

2.2 Design process

GWO algorithm imitates the hunting mode of the ground wolf, decomposes the complex problem into different subsets to produce the best solution, but the local search ability is insufficient. PIO algorithm has the advantages of simple principle, easy implementation and strong robustness, but it is easy to fall into local optimum in search. In the proposed HSGWO-MPIO algorithm, the advantages of both GWO algorithm and PIO algorithm are combined to solve the optimization problems. The GWO algorithm is simplified to explore the possible solutions available in the search space and generate offspring. Then, the modified map and compass operator phase of the PIO algorithm were performed to exploit local solutions. The procedure of the HSGWO-MPIO algorithm is given as followings.

Step 1: Initialization

In order to improve the average fitness of individuals and avoid generating infeasible solutions, a method of population initialization in a restricted area is designed. The lower boundary is the coordinate of the previous location point, and the upper boundary is the coordinate of the task target point. Therefore, in the path direction (from the starting point to the target point), the initialization distribution is basically satisfied, and the selection of task points obeys the uniform distribution in the specified interval:

$$x_{i+1} \sim U(x_i, t_x)$$
$$y_{i+1} \sim U(y_i, t_y)$$

where x_i, x_{i+1} are the X coordinate; y_i, y_{i+1} are the Y coordinate. t_x and t_y are the X and Y coordinates of the target point, respectively. U is the uniform distribution function of task points.

Step 2: Updating individual position with SGWO

In the social hierarchy stage of the GWO algorithm, the three wolves with the best fitness in the wolf group are marked as α, β, δ. The optimization process of the GWO algorithm is mainly guided by the three best solutions in each generation of population. In order to avoid the interference of too many candidate solutions to the optimization process of the algorithm, we simplified the GWO algorithm. We reserved the wolf α with the best fitness in the simplified GWO algorithm, and the update position of α wolf is selected as the optimal solution.

To accelerate the convergence speed, explore the solutions influenced by the best wolf in Eq. (4), individuals update their positions only affected by α wolf shown in Eqs. (4) and (5). Through this change, the advantage of the GWO algorithm in the exploration can be retained:

$$a(I) = (a_f - a_s)\left(1 - \frac{I}{I_{\max}}\right) \tag{1}$$

$$A(I) = a(2r - 1) \tag{2}$$

$$C = 2r \tag{3}$$

$$D(I) = |CX_\alpha(I) - X_i(I)| \tag{4}$$

$$X_i(I + 1) = X_\alpha(I) - A(I)D(I) \tag{5}$$

where a_s and a_f are the initial and stop values of the attenuation factor, I is the current iteration, and $I_{\max}$ is the maximum value of the iteration, r is the random factor of the range [0, 1], and $A(I)$ and C represent synergy coefficients. We know that the number of iterations I increases linearly. This feature can effectively ensure the balance between global search and local search.

Step 3: Updating individual position with MPIO

To improve the exploitation ability and avoid generating infeasible solutions, the modified map and compass operator phase of PIO algorithm were proposed for ensuring the efficiency of the proposed hybrid algorithm. Firstly, the general direction is identified by the geomagnetic field, and then, the current method is modified by the landform scene. The modified map and compass operator phase are as follows:

$$V_i(I) = V_i(I - 1)e^{-RI} + r(X_\alpha(I) - X_i(I - 1)) \tag{6}$$

$$X_i(I) = X_i(I - 1) + \omega V_i(I) \tag{7}$$

where R represents the map factor; generally, 0.2. ω represents the inertia factor with a value range of [0, 1].

In the search formula of modified PIO algorithm, firstly, the general direction is identified by the geomagnetic field, and then, the current method is modified by the landform scene. At the same time, adjust the direction and position of the best pigeons until they reach the exact destination. The iterative search position of wolf α is the best position, so the speed formula is improved to $V_i(I) = V_i(I-1)e^{-RI} + r\left(X_\alpha(I) - X_i(I-1)\right)$, and the linear decreasing inertia weight factor ω is added to the search position, in order to avoid falling into precocity in the search stage.

Step 4: Update the HSGWO-MPIO algorithm location

After reserving the best individual location and updating the individual location, the algorithm search location update formula is as follows:

$$X_{\text{inew}}(I+1) = X_\alpha(I) - A(I)D(I) + \omega V_i(I) \tag{8}$$

By adding a linear inertia weight factor of ω to the search location update formula, it helps the algorithm carry out dynamic optimization and avoid falling into a local optimal solution during the search phase.

Step 5: Go to Step2 until the termination condition is met

If the number of iterations reaches the maximum number, or the optimum solution has been found, the algorithm stops, otherwise, the algorithm continues to search for optimum result. If the standard deviation of the algorithm is $s < 0.3$, it can be judged that the algorithm reaches the optimal solution. The flowchart of the HSGWO-MPIO algorithm is shown in Fig. 1, and the algorithm pseudocode is shown in Algorithm 1.

Algorithm 1 The HSGWO-MPIO algorithm

```
1: Set the parameters, calculate the population initialization range
2: I = 0
3: Initialize population position
4: Calculate the fitness of each individual
5: X_α = the best individual
Optimize
6: While I < I_max or s ≥ 0.3 do
7: Calculate a, A, C
8:     for i = 1 to N
9:         Update the position of X_i(I) by the Eq.(5)
10:        Calculate X_inew(I) by the Eq.(8)
11:            if X_inew(I) fitter than X_i(I) then
12:                Update X_i(I) with X_inew(I)
13: End if
14: End for
15: I = I + 1
16: End while
Return results
Terminate
```

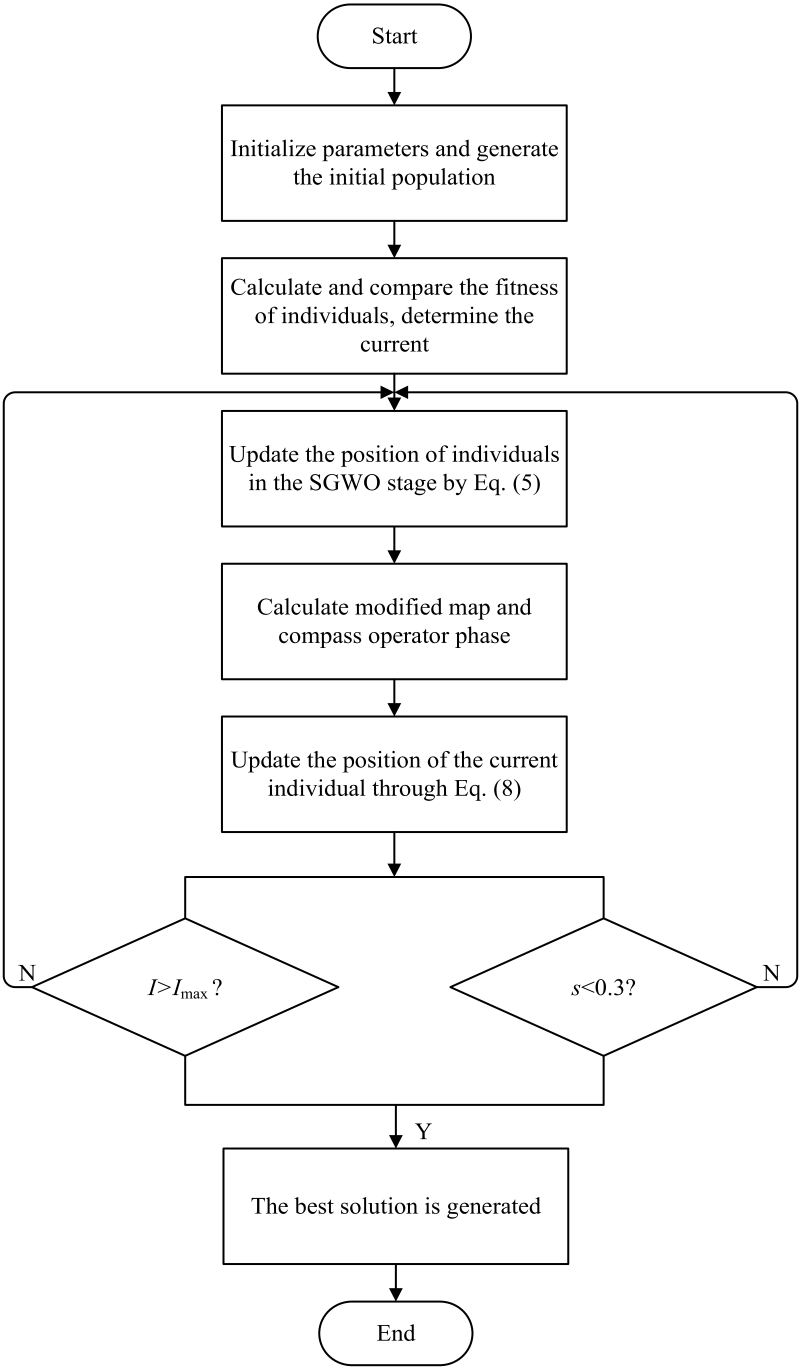


Fig. 1 Flowchart of the HSGWO-MPIO algorithm

Remark 1 The proposed HSGWO-MPIO algorithm improves the performance of the intelligent optimization algorithm in terms of convergence speed and convergence accuracy. It balances the global search and local search capabilities and lays a good foundation for practical applications.

2.3 Convergence analysis

After describing the HSGWO-MPIO algorithm design process, we will do the convergence analysis in this subsection.

Definition 1 [32] Linear Difference Equations. Define $B(\cdot) : J^+ \to R^n$ is a square nonsingular matrix,J^+ is a set of the nonnegative integer, $f^*(\cdot) : J^+ \to R^n$,$X(I) \in R^n$, for each $I \in J^+$, the linear difference equation and corresponding homogeneous linear equation with variable coefficients are as follows:

$$\begin{aligned} X(I+1) &= B(I)X(I) + f(I) \\ X(I+1) &= B(I)X(I) \end{aligned}$$

Theorem 1 *The position update equation of the HSGWO-PIO algorithm is a first-order linear time-varying difference equation.*

Proof According to Eqs. (1)–(8), the whole position update equation of the HSGWO-MPIO algorithm is:

$$\begin{aligned} X_{\text{inew}}(I+1) &= X_\alpha(I) - CA(I)X_\alpha(I) + A(I)X_i(I) + +\omega V_i(I-1)e^{-RI} \\ &\quad + r\omega X_\alpha(I) - r\omega X_i(I-1) \\ &= A(I)X_i(I) + [1 - CA(I) + r\omega]X_\alpha(I) + \omega V_i(I-1)e^{-RI} \\ &\quad - r\omega X_i(I-1) \end{aligned} \tag{9}$$

where $X_\alpha(I)$ means the best individual in the current iteration. Define:

$$B(I) = A(I) \tag{10}$$

$$f(I) = [1 - CA(I) + r\omega]X_\alpha(I) + \omega V_i(I-1)e^{-RI} - r\omega X_i(I-1)$$

Thus, Eq. (9) can be written as follows:

$$X(I+1) = B(I)X(I) + f(I)$$

So the position update equation of the HSGWO-MPIO algorithm can be written as the form of a first-order linear time-varying difference equation, and Theorem 1 is proved. □

Remark 2 When $D(I) = X_i(I) - CX_\alpha(I)$, define $B(I) = -A(I)$, $f(I) = [1 + CA(I) + r\omega]X_\alpha(I) + \omega V_i(I-1)e^{-RI} - r\omega X_i(I-1)$. Then, we can get $X(I+1) = B(I)X(I) + f(I)$; Theorem 1 is proved.

Lemma 1 [32]

$$\forall f(I) \in [J^+, R^n],$$

the stability degree of the linear difference equation is equivalent to the stability degree of the general solution of the corresponding homogeneous linear equation.

Lemma 2 [32] *If there is a kind of matrix norm that satisfy*:

$$\|B(I)\|_X \leq \frac{I+1}{I+2}, I = I_0, I_0 + 1, \dots$$

then $X(I+1) = B(I)X(I) + f(I)$ *can be guaranteed uniformly asymptotically stability, where* $\| \cdot \|_X$ *means a certain norm,* $\forall I_0 \in R$.

Theorem 2 *When the condition of* $I \to I_{\max}$ *is met, individuals will converge to the local or global optimum solutions.*

Proof According to Eqs. (1), (2) and (10), we have:

$$\begin{aligned} \lim_{I \to I_{\max}} A(I) &= \lim_{I \to I_{\max}} (2r-1)\left[(a_f - a_s)\left(1 - \frac{I}{I_{max}}\right)\right] \\ &= (2r-1) \lim_{I \to I_{\max}} \left[(a_f - a_s)\left(1 - \frac{I}{I_{max}}\right)\right] = 0 \end{aligned} \tag{11}$$

Thus, when the condition of $I \to I_{\max}$ is met, according to Lemmas 1, 2, it can be ensured that the position update equation based on the SGWO-MPIO algorithm is stable. In other words, each individuals X_i will converge to a steady state X_e when the condition of is met($\lim_{I \to I_{\max}} X_i(I) = X_e$). So:

$$\lim_{I \to I_{\max}} X_i(I+1) = \lim_{I \to I_{\max}} X_i(I) = X_e \tag{12}$$

Substitute Eq. (9) into Eq. (12), we have:

$$\begin{aligned} \lim_{I \to I_{\max}} X_i(I+1) = \lim_{t \to I_{\max}} & X_\alpha(I) - CA(I)X_\alpha(I) + A(I)X_i(I) \\ & + \omega V_i(I-1)e^{-RI} + r\omega X_\alpha(I) - r\omega X_i(I-1) = X_e \end{aligned} \tag{13}$$

Based on the Eqs. (10), (11), (13) is formed into:

$$\begin{aligned} \lim_{I \to I_{\max}} X_i(I+1) = & \lim_{t \to I_{\max}} X_\alpha(I) - CA(I)X_\alpha(I) + A(I)X_i(I) \\ & + \omega V_i(I-1)e^{-RI} + r\omega X_\alpha(I) - r\omega X_i(I-1) \\ = & \lim_{I \to I_{\max}} X_\alpha(I) + \omega V_i(I-1)e^{-RI} \\ & + r\omega X_\alpha(I) - r\omega X_i(I-1) \\ = & X_e \end{aligned}$$

Because of the randomness of α grey, we can get that:

$$\begin{cases} \lim_{I \to I_{\max}} X_i(I) = X_\alpha \\ \lim_{I \to I_{\max}} X_i(I+1) = X_\alpha \end{cases}$$

Therefore, Eq. (12) can be written as follows:

$$\lim_{I \to I_{\max}} X_i(I+1) = \lim_{I \to I_{\max}} X_i(I) = X_e = X_\alpha$$

So, when the condition of $I \to I_{\max}$ is met, all individuals can converge to the optimum solutions, and Theorem 2 is proved. □

In summary, the convergence of the HSGWO-MPIO algorithm can be insured when the condition of is met.

Remark 3 When $D(I) = X_i(I) - CX_\alpha(I)$, according to the reasoning of Theorem 1, Remark 1, Lemmas 1 and 2, we can get $\lim_{I \to I_{\max}} X_i(I+1) = \lim_{I \to I_{\max}} X_i(I) = X_e = X_\alpha$. Theorem 2 is proved.

Remark 4 I and $I_{\max}$ are values in the range of $[0, N^+]$. If $I_{\max}$ is too small, $\frac{I}{I_{\max}} \to 1$. So that we can get $\lim_{I \to I_{\max}} A(I) = 0$. It can be further obtained $\lim_{I \to I_{\max}} X_i(I+1) = \lim_{I \to I_{\max}} X_i(I) = X_e = X_\alpha$. So, if $I_{\max}$ is too small, all individuals can converge to the optimum solutions.

2.4 Computational complexity

As shown in the loop diagram, the algorithm can be divided into four stages: initialization, SGWO algorithm, MPIO algorithm and update location formula. The algorithm is suitable for the population with size N, the position of individual in the population is a vector with size P, and the number of iterations to task is I. The space complexity and time complexity of the algorithm are mainly affected by the phase of the algorithm.

Phase 1: Initialization

The first section of the program is executed once at the beginning to prepare for the next stage. The space complexity of this stage is $O(N * P)$. Time complexity is $O(N * P * I)$, and the complexity of the decision stop standard is $O(1)$.

Phase 2: Simplified GWO position update operator

SGWO creates a new location for each individual affected by the α wolf, and all the original locations will be replaced by the new location. The space complexity of this phase is $O\left(NX_\alpha - ADN\right)$. Time complexity is $O\left(NX_\alpha I - ADNI\right)$.

Phase 3: Modified PIO map operator

MPIO calculates the new location affected by random factors. If the new location provides a more appropriate value, then, the original location will be replaced. The space complexity of this stage is $O\left(NX_i + \omega NV_i\right)$. Time complexity is $O\left(NX_iI + \omega NV_iI\right)$.

Phase 4: Update location formula optimization stage

The updated location formula retains the essence of the GWO algorithm and the PIO algorithm and generates a new location update formula after considering the optimization efficiency and practical operation. At this stage, the space complexity is $O\left(NX_\alpha - ADN + \omega NV\right)$. The time complexity at this stage is $O\left(NX_\alpha I - ADNI + \omega NVI\right)$.

Remark 5 This part analyzes the computational complexity of each part of the HSGWO-MPIO algorithm. Compared with the GWO [22] algorithm and the PIO [31] algorithm, it has the advantages of speed and accuracy in each iteration and can effectively improve the convergence speed compared with the original algorithm.

2.5 Algorithm comparison and accuracy

Compared with traditional heuristic optimization algorithms, such as PSO algorithm [33] and WOA [34], the HSGWO-MPIO optimization algorithm can take into account the global development ability and local search ability, make up for the shortcomings of the single optimization algorithm in the convergence speed and calculation accuracy and achieve the complementary advantages between the algorithms.

Compared with the existing improved optimization algorithms, such as hybrid algorithm combing genetic algorithm with evolution strategy (GAES) [7], an improved hybrid grey wolf optimization algorithm (GWOPSO) [21] and discrete hybrid cuckoo search and simulated annealing Algorithm (DCSA) [9], the HSGWO-MPIO algorithm has advantages in principle, parameter selection, robustness, etc. The HSGWO-MPIO algorithm preserves the parts of the basic algorithm that has good optimization results, simplifies the steps of the hybrid algorithm and has advantages in algorithm principle. Moreover, in terms of parameter selection, the HSGWO-MPIO algorithm reduces the range of parameter selection. It avoids previous empirical selection and effectively improves the quality of the optimal solution. Finally, the designed HSGWO-MPIO algorithm provides a condition $s < 0.3$ for outputting the optimal solution. In this way, it improves the computational accuracy of the algorithm and ensures its robustness.

Compared with GWO [19] algorithm, HSGWO-MPIO algorithm is helpful for the algorithm to jump out of the local and maintain the development ability, improve the search ability. Compared with PIO [28] algorithm, HSGWO-MPIO algorithm is helpful to expand the candidate solution space, improve the search range and accelerate the convergence rate of the algorithm.

From the perspective of momentum, the current speed is affected by the previous speed and the optimal global position, and $0 < r < 1$ can limit the speed $V_i(I)$ to be too large. In other words, the current velocity is gradual, not transient, and it is a process of momentum. This ensures the stability and accuracy of the search, reduces the shock and reaches the optimal value quickly.

Remark 6 Since the HSGWO-MPIO algorithm is improved on the basis of the GWO algorithm and PIO algorithm. Compared with the basic optimization algorithm, the algorithm has updated the form of the attenuation factor, introduced the relationship between the global search and the local search of inertial silver balance and can quickly reach the ideal convergence state under dynamic conditions. There are obvious advantages.

3 Compared simulations

In this section, five simulation situations are considered, namely the influence of inertia factor ω on the algorithm, the influence of random factor r on the algorithm, the comparison of different algorithms, application of UAV path trajectory and ablation experiment. Given the initial range of the algorithm, the simulation of each part of the algorithm is as follows:

Simulation 1: Influence of inertia factor ω on algorithm

Make $r = 0.4$, to test the influence of inertia factor ω on the optimization ability, the following six cases based on $r = 0.4$ are simulated: $\omega = 0.4, \omega = 0.45, \omega = 0.5, \omega = 0.55, \omega = 0.6$ and $\omega = 0.65$. Simulation result is presented in Fig. 2. As can be seen from Fig. 2a, when $\omega = 0.5$, the optimization ability of the hybrid algorithm is best, which can help the algorithm effectively improve the convergence speed between global search and local search, followed by $\omega = 0.55$. When $\omega = 0.6$, $\omega = 0.65$ and $\omega = 0.4$, the convergence speed of the algorithm is obviously inferior to that of $\omega = 0.5$ and $\omega = 0.55$. Although the value of ω starts from 0, simulation results show that the optimization effect of the hybrid algorithm is not ideal when the value of $\omega < 0.4$. Figure 2b gives that the random value from 0 to 0.4 cannot achieve the optimization effect of the hybrid algorithm. Therefore, the random factor has an impact on the global search. The larger the value of ω, the stronger the global optimization ability, and vice versa.

Simulation 2: Influence of random factors r on algorithm

Make $\omega = 0.5$, to test the influence of the random factor r on the optimization ability, the following nine cases based on are simulated: $r = 0$, $r = 0.1$, $r = 0.2$, $r = 0.4$, $r = 0.5$, $r = 0.6$, $r = 0.8$, $r = 0.9$ and $r = 1$. When other variables of the algorithm remain unchanged, we change the value of r and run the algorithm to compare the optimization ability under different r. Figure 3 presents that when the value of r fluctuates around 0.4, the proposed hybrid algorithm has better optimization ability, followed by $r = 0.9$. When r is 0.1, 0.6, 0.8, 1, the convergence effect of the optimization algorithm deviates from the expectation and does not achieve the desired effect. When $r = 0.2$ and $r = 0.5$, the optimization ability of the proposed HSGWO-MPIO algorithm is worst. When $r = 0$, the algorithm cannot operate and does not reach the ability of global search and local search.

Simulation 3: Performance comparison of different algorithms

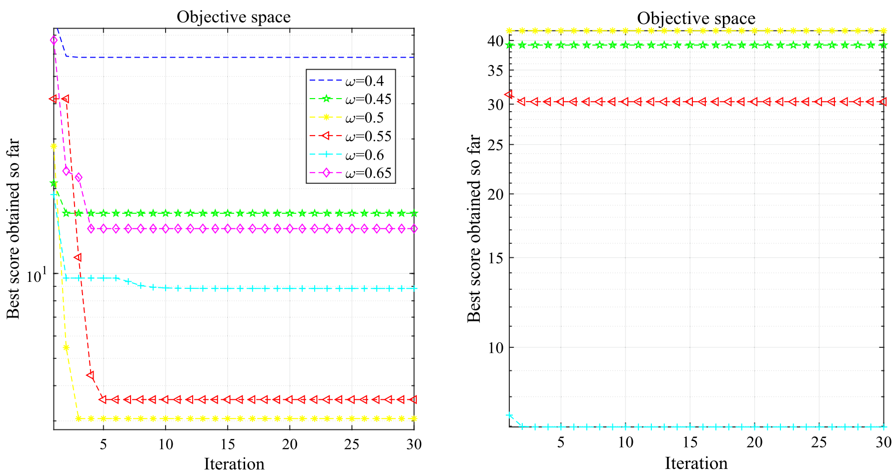
In order to study the advantage of the proposed algorithm, PSO [33], GWO [19], PIO [28], WOA [34], GAES [7], DGWO [20], GWOPSO [21] and DCSA [9] are compared with the proposed algorithm. Under the same function model, each algorithm performs 30 searches to calculate the best location. The results are shown in Table 1 and Fig. 4. Compared with the other eight algorithms, the simulation result of the proposed algorithm has smaller mean values. Meanwhile, a lower standard deviation value proves that the HSGWO-MPIO algorithm can search for the optimal path stably. The optimal search scores of PSO, GWO, PIO, WOA, GAES, DGWO, GWOPSO and DCSA are 3.437, 9.138, 3.651, 3.136, 3.434, 3.197, 8.498 and 3.013, respectively. Our algorithm has the best stability among all comparison algorithms, and its convergence rate is shown in Fig. 5. It can be clearly seen from the figure that the designed HSGWO-MPIO algorithm has the fastest convergence rate of all comparison algorithms.

Simulation 4: Comparison of UAV trajectory application based on different algorithms

In this experiment, the path planning diagram of UAV is generated under GWO algorithm, PIO algorithm and HSGWO-MPIO algorithm, respectively. The objective function and constraint conditions of UAV flight path are as follows:

$$\min\ J = \sum_{i=1}^{l} \left(k_1 L_i^2 + k_2 H_i^2 + k_3 T_i^2\right)$$
$$s.t.\ d_f - R_j > 0$$

where l represents the number of track segments, L_i represents the track length of segment i, this item represents the distance cost, H_i represents the average altitude of segment i, this item represents the altitude cost, T_i represents the threat index of segment i, and this item represents the threat cost. $k_1 + k_2 + k_3 = 1$ represents the weight value of distance cost, altitude cost and threat cost, and the selection of



(a) (b)

Fig. 2 Influence of inertia factor ω on algorithm

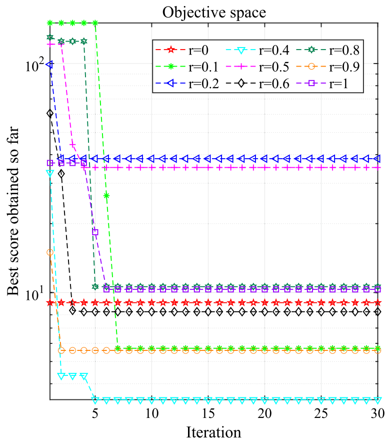


Fig. 3 *r* parameter diagram

weight value is related to the flight mission requirements. d_f represents the UAV fuselage width, and R_j represents the threat radius.

The whole space is set into three random obstacle avoidance areas. The starting coordinates of the UAV are [0, 0] km and the coordinates of the target point are [4, 6] km. The speed range of UAV is [100, 200] m/s. Other parameters are as follows: the population was 100, the number of iterations was 50, $a_f = 0$, $a_2 = 2$, $r = 0.42$, $\omega = 0.5$. Under the same model and parameter selection, the UAV flight trajectory figures using GWO algorithm, PIO algorithm and HSGWO-MPIO algorithm are compared. In Fig. 6, it can be seen that compared with the proposed hybrid algorithm, the flight trajectory using a single algorithm can generate a smoother flight path but cannot successfully avoid all threats, while the HSGWO-MPIO algorithm can avoid threats and generate smooth flight trajectory.

Simulation 5: Ablation experiment

In Simulation 1 and Simulation 2, we have discussed the influence of random factor r and inertia factor ω on the convergence speed of the algorithm under different

Table 1 Comparison of search location values

Algorithm name	Worst score	Best score	Median value	Mean value	Standard deviation
PSO	5.491	3.437	4.982	4.464	1.148
GWO	12.54	9.138	11.01	10.839	1.902
PIO	6.175	3.651	4.806	4.913	1.411
WOA	6.089	3.136	4.517	4.6125	1.651
GAES	4.443	3.434	4.03	3.9385	0.564
DGWO	4.304	3.197	3.701	3.7505	0.6188
GWOPSO	11.21	8.498	10.25	9.854	1.416
DCSA	3.281	3.013	3.115	3.147	0.2598
HSGWO-MPIO	3.136	3.005	3.083	3.0705	0.173

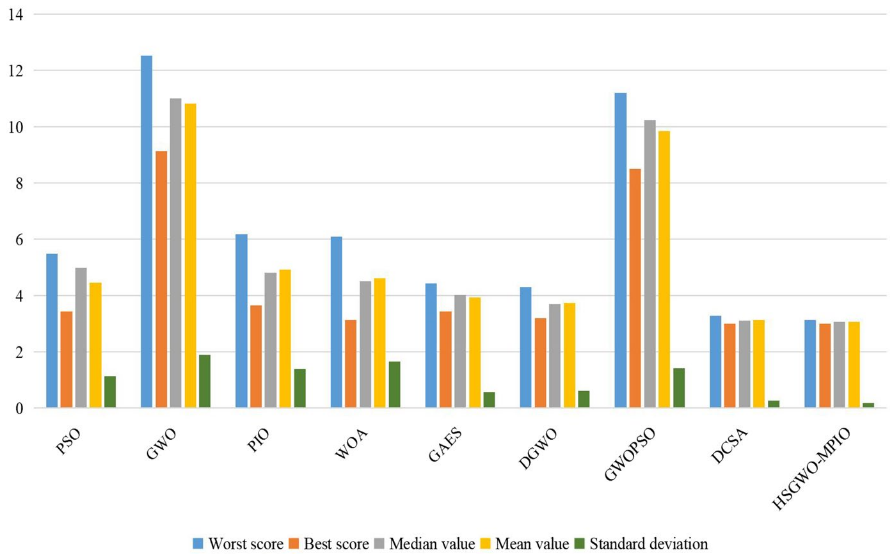
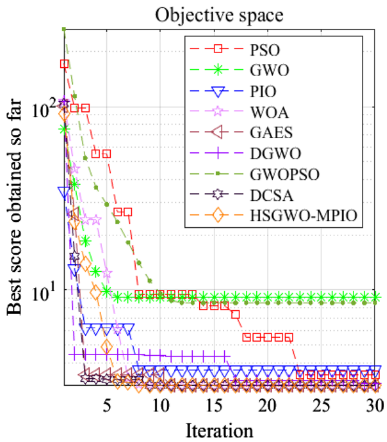


Fig. 4 Standard map of algorithm comparison experiment

Fig. 5 The convergence of compared algorithms



values. The results show that when $r = 0.4$ and $\omega = 0.5$, the convergence performance of the algorithm is the best.

In order to further verify the performance of the optimization algorithm in practical applications, in this experiment, we discuss the influence of inertia factor ω and random factor r on the flight path of UAV under different values.

Case 1: Influence of inertia factor ω on UAV flight trajectory

To test the influence of inertia factor ω on UAV flight trajectory, let random factor r the following nine cases are simulated: $\omega = 0.1$, $\omega = 0.2$, $\omega = 0.4$, $\omega = 0.45$, $\omega = 0.5$, $\omega = 0.55$, $\omega = 0.6$, $\omega = 0.65$ and

$\omega = 0.8$. Simulation result is presented in Fig. 7. As can be seen from Fig. 7, when $\omega = 0.5$, UAV can efficiently avoid threats and generate optimal trajectory. When $\omega = 0.55$, the UAV optimization capability takes the second place. Through many simulation experiments, it can be found that the change of ω has an impact on the global search ability of UAV. When $\omega < 0.4$, the UAV cannot successfully avoid the threat. When $\omega \geq 0.4$, the UAV can successfully avoid the threat and generate flight path. It can be seen from the figure that inertia factor ω focuses on the global search, which changes the search weight of the compass operator. This is conducive to the algorithm jumping out of the local, and the effect is obvious in the later stage of iteration.

Case 2: Influence of random factor r on UAV flight trajectory

To test the influence of random factor r on UAV flight trajectory, let inertia factor $\omega = 0.5$ the following nine cases are simulated: $r = 0$, $r = 0.1$, $r = 0.2$, $r = 0.4$, $r = 0.5$, $r = 0.6$, $r = 0.8$, $r = 0.9$ and $r = 1$. Simulation result is presented in Fig. 8. As can be seen from Fig. 8, when $r = 0.4$, UAV can efficiently avoid threats and generate optimal trajectory. When $r = 0.9$, the UAV optimization capability takes the second place. With the same other parameters, it can be seen from Fig. 8 that when r takes other values, the safe distance between the UAV and the threat is too small during flight, which is not conducive to the flight mission. Through many simulation experiments, it can be seen that the change of r affects the local search ability of UAV. The change of the random factor r values will cause the values of A to change, which determines whether to search for optimization in various regions or to search for some regions in a centralized way. This is conducive to the identification and avoidance of threats, which is crucial at every step of the iteration.

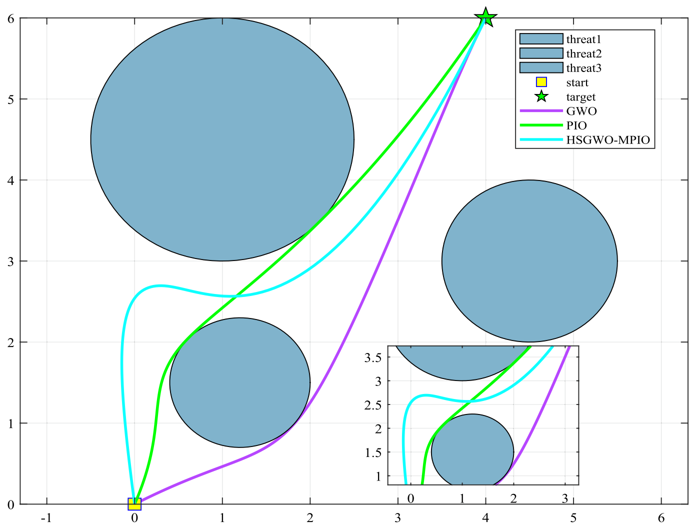


Fig. 6 Flight trajectory of UAV under three algorithms

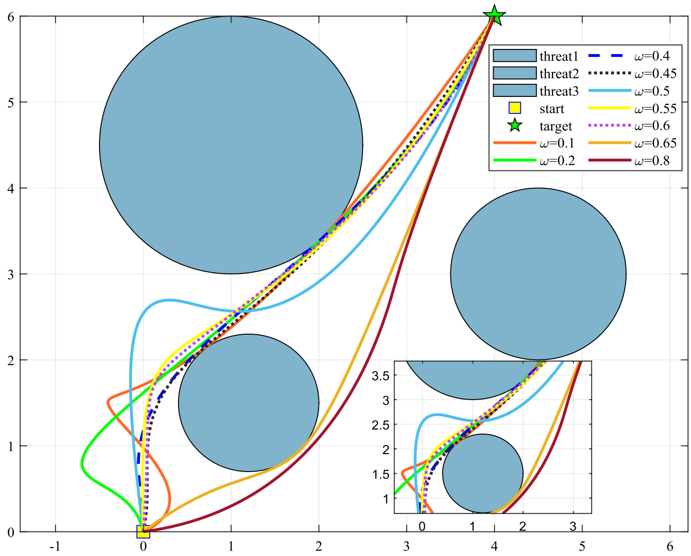


Fig. 7 Influence of inertia factor ω on UAV flight trajectory

4 Conclusions

This paper proposes the HSGWO-MPIO algorithm based on simplified GWO algorithm and modified PIO algorithm. The algorithm effectively combines the exploration ability of GWO algorithm with the development ability of PIO algorithm. The ability of population initialization is enhanced, the convergence stage of GWO algorithm is simplified, and a new decay factor is designed to maintain the search

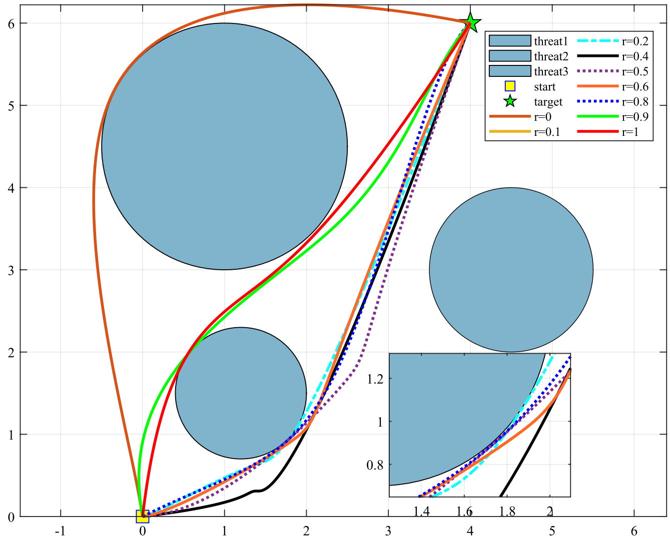


Fig. 8 Influence of random factor r on UAV flight trajectory

ability of the population. The map and compass operator phase of PIO algorithm were modified, and a new position update formula is designed to improve the development ability of the algorithm. Then, the convergence of HSGWO-MPIO algorithm is analyzed by using the linear difference equation method. At the same time, the computational complexity is analyzed by the population size and dimension. Finally, comparative experimental results show that HSGWO-MPIO algorithm has better convergence speed and search accuracy than those of other heuristic optimization algorithms. The HSGWO-MPIO algorithm can successfully obtain an effective and safe flight path. In the future, we will be devoted to reducing the complexity of hybrid algorithms and conducting research on high-dimensional optimization algorithms. In terms of application, the HSGWO-MPIO algorithm will be applied to the path planning of multiple UAVs, providing better service support for combat scenarios, express delivery scenarios and agricultural scenarios.

Acknowledgements All authors contributed to the study conception and design. The conceptualization and the methodology were formulated by XZ and FW. Data curation was completed by XZ and RS. And the formal analysis was finished by XZ, FW and CZ. All authors have read and agreed to the published version of the manuscript.

Funding This work was supported in part by Science and Technology Project of Hebei Education Department (ZD2022012), Natural Science Foundation of Hebei Province (F2020203105, F2022203085) and National Natural Science Foundation of China (62073234).

Data availability statement The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Conflict of interest The authors have no conflict of interest.

References

1. Jin Z, Shima T, Schumacher CJ (2006) Optimal scheduling for refueling multiple autonomous aerial vehicles. IEEE Trans Robot 22(4):682–693. https://doi.org/10.1109/TRO.2006.878793
2. She R, Ouyang Y (2020) Efficiency of UAV-based last-mile delivery under congestion in low-altitude air. Transp Res C 122:1–13. https://doi.org/10.1016/j.trc.2020.102878
3. Pardo D, Möller L, Neunert M et al (2016) Evaluating direct transcription and nonlinear optimization methods for robot motion planning. IEEE Robot Autom Lett 1(2):946–953. https://doi.org/10.1109/LRA.2016.2527062
4. Han Y, Wang B, Deng Z et al (2016) An improved TERCOM-based algorithm for gravity-aided navigation. IEEE Sens J 16(8):2537–2544. https://doi.org/10.1109/JSEN.2016.2518686
5. Zhu Z, Yin Y, Lyu H (2023) Automatic collision avoidance algorithm based on route-plan-guided artificial potential field method. Ocean Eng 271:1–23. https://doi.org/10.1016/j.oceaneng.2023.113737
6. Bertsimas D, Tsitsiklis J (1993) Simulated annealing. Stat Sci 8(1):10–15. https://doi.org/10.1214/ss/1177011077

7. Choi K, Jang DH, Kang SI et al (2016) Hybrid algorithm combing genetic algorithm with evolution strategy for antenna design. IEEE Trans Magn 52(3):1–4. https://doi.org/10.1109/TMAG.2015.2486043
8. Yu X, Li C, Zhou J (2020) A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios. Knowl Based Syst 204:1–11. https://doi.org/10.1016/j.knosys.2020.106209
9. Alkhateeb F, Abed-alguni BH, Al-rousan MH (2022) Discrete hybrid cuckoo search and simulated annealing algorithm for solving the job shop scheduling problem. J Supercomput 78:4799-C4826. https://doi.org/10.1007/s11227-021-04050-6
10. Abdeyazdan M (2017) A new method for the informed discovery of resources in the grid system using particle swarm optimization algorithm. J Supercomput 73:5354-C5377. https://doi.org/10.1007/s11227-017-2090-y
11. Yan F (2020) Gauss interference ant colony algorithm-based optimization of UAV mission planning. J Supercomput 76:1170-C1179. https://doi.org/10.1007/s11227-018-2540-1
12. Lee JH, Song JY, Kim DW et al (2018) Particle swarm optimization algorithm with intelligent particle number control for optimal design of electric machines. IEEE Trans Ind Electron 65(2):1791–1798. https://doi.org/10.1109/TIE.2017.2760838
13. Li H, Baoyin H (2020) Optimization of multiple debris removal missions using an evolving elitist club algorithm. IEEE Trans Aerosp Electron Syst 56(1):773–784. https://doi.org/10.1109/TAES.2019.2934373
14. Cheng R, Song Y, Chen D et al (2019) Intelligent positioning approach for high speed trains based on ant colony optimization and machine learning algorithms. IEEE Trans Intell Transp Syst 20(10):3737–3746. https://doi.org/10.1109/TITS.2018.2878442
15. Wang Y, Zhang Y, Xu D et al (2022) Improved whale optimization-based parameter identification algorithm for dynamic deformation of large ships. Ocean Eng 245:1–9. https://doi.org/10.1016/j.oceaneng.2021.110392
16. Khan MSA, Santhosh R (2021) Task scheduling in cloud computing using hybrid optimization algorithm. Soft Comput. https://doi.org/10.1007/s00500-021-06488-5
17. Kumar RP, Raj JS, Smys S (2021) Performance analysis of hybrid optimization algorithm for virtual head selection in wireless sensor networks. Wirel Pers Commun 123:1925–1940. https://doi.org/10.1007/s11277-021-09222-4
18. Chandrasekhar U, Khare N (2021) An intelligent tutoring system for new student model using fuzzy soft set-based hybrid optimization algorithm. Soft Comput 25:14979–14992. https://doi.org/10.1007/s00500-021-06396-8
19. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
20. Pan C, Si Z, Du X et al (2021) A four-step decision-making grey wolf optimization algorithm. Soft Comput 25:14375–14391. https://doi.org/10.1007/s00500-021-06194-2
21. Teng Z, Lv J, Guo L (2019) An improved hybrid grey wolf optimization algorithm. Soft Comput 23:6617–6631. https://doi.org/10.1007/s00500-018-3310-y
22. Zhao Y, Li W, Liu A (2020) Improved grey wolf optimization based on the two-stage search of hybrid CMA-ES. Soft Comput 24:1097–1115. https://doi.org/10.1007/s00500C019C03948Cx
23. Dev K, Poluru RK, Kumar L et al (2021) Optimal radius for enhanced lifetime in IoT using hybridization of rider and grey wolf optimization. IEEE Trans Green Commun Netw 5(2):635–644. https://doi.org/10.1109/TGCN.2021.3069187
24. Qu C, Gai W, Zhang J et al (2020) A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning. Knowl Based Syst 194:1–14. https://doi.org/10.1016/j.knosys.2020.105530
25. Pradhan M, Roy PK, Pal T (2016) Grey wolf optimization applied to economic load dispatch problems. Int J Electr Power Energy Syst 83:325–334. https://doi.org/10.1016/j.ijepes.2016.04.034
26. Yao P, Wang H, Ji H (2016) Multi-UAVs tracking target in urban environment by model predictive control and improved grey wolf optimizer. Aerosp Sci Technol 55:131–143. https://doi.org/10.1016/j.ast.2016.05.016
27. Shakarami M, Davoudkhani IF (2016) Wide-area power system stabilizer design based on grey wolf optimization algorithm considering the time delay. Electr Power Syst Res 133:149–159. https://doi.org/10.1016/j.epsr.2015.12.019

28. Duan H, Qiao P (2014) Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. Int J Intell 7(1):24–37. https://doi.org/10.1108/IJICC-02-2014-0005
29. Duan H, Huo M, Shi Y (2020) Limit-cycle-based mutant multiobjective pigeon-inspired optimization. IEEE Trans Evol Comput 24(5):948–959. https://doi.org/10.1109/TEVC.2020.2983311
30. Duan H, Zhao J, Deng Y et al (2021) Dynamic Discrete Pigeon-Inspired Optimization for Multi-UAV Cooperative Search-Attack Mission Planning. IEEE T Aero Elec Sys 57(1):706–720. https://doi.org/10.1109/TAES.2020.3029624
31. Wu Z, Liu Y (2022) Integrated optimization design using improved pigeon-inspired algorithm for a hypersonic vehicle model. Int J Aeronaut Space 23(5):1033–1042. https://doi.org/10.1007/s42405-022-00492-1
32. Chen Y, Yu J, Mei Y et al (2016) Modified central force optimization (MCFO) algorithm for 3D UAV path planning. Neurocomputing 171:878–888. https://doi.org/10.1016/j.neucom.2015.07.044
33. James K, Russell E (1995) Particle swarm optimization. In: Proceedings of ICNN'95: International Conference on Neural Networks, vol 4, pp 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
34. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008