A Novel Algorithm for Robot Path Planning Based on Hybrid Strategy Improved Wild Horse Optimizer

Juntao Zhao¹, Xiaochuan Luo¹, and Yong Li¹

¹Northeastern University College of Information Science and Engineering

August 21, 2023

Abstract

In order to solve the path planning problem of mobile robots, a novel hybrid strategy improved wild horse optimizer (HI-WHO) is proposed in this paper. The algorithm utilizes Sobol sequence to initialize the population, which ensures a uniform distribution of initial solutions in the search space. It also integrates Lévy flight strategy and nonlinear dynamic adaptive factor to balance the exploration and exploitation ability and improve the search efficiency and quality at different stages of the algorithm. In addition, the algorithm can ensure the global search capability and the ability to jump out of the local optimum by using the lens imaging opposition-learning strategy and greedy mechanism. In the simulation experiments, 20 benchmark functions are selected to verify the effectiveness of the proposed method. Finally, the improved algorithm, combined with cubic B-Spline interpolation, solves the path planning problem by establishing the mathematical model for mobile robot path planning, and the performance is evaluated by conducting simulation experiments in a simple experimental environment and a real environment with grid map. The results show that the HI-WHO has good stability and optimal comprehensive performance, showcasing its effectiveness in addressing the robot path planning problem.
A Novel Algorithm for Robot Path Planning Based on Hybrid Strategy Improved Wild Horse Optimizer

Juntao Zhao¹² | Xiaochuan Luo*¹³ | Yong Li¹

¹College of Information Science and Engineering, Northeastern University, Shenyang, China
²EPROAD EA 4669, University of Picardy Jules Verne, Amiens, France
³State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, China

Abstract

In order to solve the path planning problem of mobile robots, a novel hybrid strategy improved wild horse optimizer (HI-WHO) is proposed in this paper. The algorithm utilizes Sobol sequence to initialize the population, which ensures a uniform distribution of initial solutions in the search space. It also integrates Lévy flight strategy and nonlinear dynamic adaptive factor to balance the exploration and exploitation ability and improve the search efficiency and quality at different stages of the algorithm. In addition, the algorithm can ensure the global search capability and the ability to jump out of the local optimum by using the lens imaging opposition-learning strategy and greedy mechanism. In the simulation experiments, 20 benchmark functions are selected to verify the effectiveness of the proposed method. Finally, the improved algorithm, combined with cubic B-Spline interpolation, solves the path planning problem by establishing the mathematical model for mobile robot path planning, and the performance is evaluated by conducting simulation experiments in a simple experimental environment and a real environment with grid map. The results show that the HI-WHO has good stability and optimal comprehensive performance, showcasing its effectiveness in addressing the robot path planning problem.

KEYWORDS
Wild horse optimizer(WHO), Sobol sequence, Lévy flight, dynamic self-adaptive factor, opposition-based learning, cubic B-Spline, path planning

1 | INTRODUCTION

In recent years, the rapid advancement of robot technology has led to a substantial increase in the demand for autonomous robots. As a vital component of autonomous systems, robot path planning has garnered significant attention from researchers (Lin, Liu, Wang, & Kong, 2023; Houacine & Drias, 2021; Su, Ju, Xu, & Dai, 2023). The primary objective of robot path planning is determining the optimal trajectory for a robot to navigate from a given starting point to a designated destination within a specific environment. This trajectory must not only minimize the traversal time and distance but also ensure collision avoidance with obstacles while adhering to defined constraints (Ntakolia, Moustakidis, & Siouras, 2023).

The problem of robot path planning is a challenging NP-hard problem. Currently, the path planning methods can be categorized into two types: classical methods and metaheuristic algorithms (Luan & Thin, 2023). Classical optimization algorithms such as gradient descent algorithm and genetic algorithm often suffer from problems like getting stuck to a local optima, slow convergence speed, and low solution quality, significantly affecting the accuracy and efficiency of path planning. To address these challenges, researchers have proposed various metaheuristic algorithms, such as Particle Swarm Optimization (PSO) (L. Xu, Cao, & Song, 2022), Artificial Bee Colony (ABC) (Kumar & Sikander, 2022), Harris Hawks Optimization (HHO) (Alweshah et al., 2022), and Sparrow Search Algorithm (SSA) (G. Zhang & Zhang, 2022a), which have been applied to solve robot path planning problems. Metaheuristic algorithms offer advantages such as few adjustable parameters, no gradient mechanisms, strong parallelism, and ease of interpretation and understanding, making them widely utilized in mobile robot path planning problems.

Whereas, when metaheuristic algorithms are applied to solve path planning problems in complex environments, they often suffer from drawbacks such as slow convergence speed, low optimization accuracy, and a tendency to get trapped in local
optima, primarily due to a reduction in population diversity in later stages \cite{Dai, Yu, Zhang, Zhan, & Zheng, 2023}. To overcome these limitations, researchers have made efforts to combine metaheuristic algorithms with traditional optimization algorithms or apply various strategies to improve the performance. A bi-level multi-objective path planning algorithm for Autonomous Underwater Vehicles (AUVs) is proposed in \cite{Sui et al., 2023}, which integrates Ant Colony Optimization (ACO), Particle Swarm Optimization, and A* algorithm to obtain the optimal collision-free path. The algorithm iteratively updates the ACO using feedback from the inner layer, resulting in the shortest collision-free path traversing all tasks. In \cite{Z. Zhang, He, & Yang, 2022}, an improved Sparrow Search Algorithm is proposed, which incorporates strategies such as linear path fusion, improved neighborhood search, and position update functions. This algorithm addresses the problems of jaggedness in the planned path corners and slow convergence speed. A hybrid algorithm is developed in \cite{Sahu, Das, & Kumar, 2023} by efficiently combining modified cuckoo search, sine cosine algorithm, and particle swarm optimization. The proposed approach can compute an optimal collision-avoiding path for a stick-carrying twin, aiming to move from a predefined start position to a designated goal position. By incorporating these techniques, researchers have achieved notable improvements in the performance of metaheuristic algorithms for path planning problems in complex environments.

Similar to the metaheuristic algorithms mentioned above, the Wild Horse Optimizer (WHO) is a newly developed optimization method that demonstrates remarkable performance in solving complex optimization problems \cite{Naruei & Keynia, 2022}. However, the WHO still has several limitations, such as insufficient population diversity during initialization, limited global search capability, and difficulties in escaping local optima in the later stages of the algorithm. These factors result in low search accuracy for multi-peak optimization problems and make it prone to getting trapped in local optima. Additionally, the algorithm struggles to effectively utilize individual information for achieving better results \cite{Zheng et al., 2022}. Numerous researchers have conducted relevant studies to improve its effectiveness to enhance the overall performance of the Wild Horse Optimizer (WHO). The study in \cite{Vasanthkumar et al., 2022} applies the improved Wild Horse Optimizer (IWHO) as a hyperparameter optimizer for the deep learning enabled battery management system (IWHODL-BMS) in the Internet of Things (IoT) based hybrid electric vehicles (HEVs). The proposed improved Wild Horse Optimizer (I-WHO) in \cite{Sayed & Hassanien, 2022} is applied to enhance the performance of solving global and combinatorial optimization problems.

Once path points are generated using a metaheuristic algorithm based on specific criteria, it is necessary to establish connections between these points using line segments while also ensuring the smoothness of the path to adhere to the kinematic constraints of the robot \cite{Luan & Thinh, 2023, Gao, Zhou, Zhao, & Shao, 2023}. The smooth path is crucial to prevent frequent stops and rapid turns of mobile robots, which result in additional energy consumption \cite{Hu, Du, & Wei, 2023, Song, Wang, Zou, Xu, & Alsaadi, 2019}. Consequently, there has been a proliferation of research works aimed at robot path smoothing. In \cite{Singh, 2023}, a modified Smooth-Bidirectional RRT (S-BRRT) utilizes adaptive control points in the Bezier curve technique to achieve smoother trajectories. In \cite{Haghighi, Delahaye, & Asadi, 2023}, a novel hybrid optimization framework, namely the Hybrid Dubins-Simulated Annealing (HDSA) method, incorporates Dubins paths and Apollonius’ tangent line to generate trajectory candidates that adhere to the post-failure performance characteristics of the distressed aircraft. Although the Dubins curve is simple and effective for generating smooth paths, the connectivity points of its elements fail to meet certain requirements for path smoothness, such as curvature continuity. This results in the need for additional transition curves \cite{Fraichard & Scheuer, 2004}.

Based on the analysis above, to overcome the limitations of the standard WHO and further enhance its performance and applicability, this paper proposes a new robot path planning algorithm based on the Hybrid Improved Wild Horse Optimizer (HI-WHO). The HI-WHO combines various improvement strategies, including (i) Sobol sequence for population initialization, (ii) Lévy flight for individual position update, (iii) nonlinear dynamic adaptive factors, and lens imaging opposition-based learning strategy for optimal individual position update. These strategies aim to improve the global search capability and convergence speed of the original WHO. Additionally, a mathematical model for the mobile robot path planning problem is established, and the planned path is further smoothed using cubic B-Spline curves to make it more suitable for robot navigation and motion control. The effectiveness and stability of the improved algorithm is verified by performing robot path planning experiments in simple and real environment grid maps. The contributions of this research work can be summarized as follows:

1. Enhancement of the original WHO through integrating hybrid strategies, which leads to improved optimization accuracy and ability to escape local optima. The proposed HI-WHO demonstrates stronger global search capabilities and faster convergence speed.

2. Development of a mathematical model for robot path planning problems and solve it utilizing the improved HI-WHO in conjunction with cubic B-Spline curves. This approach can generate smooth and optimized paths for robot navigation and motion control.
3. Expansion of the application scope of the WHO by demonstrating its effectiveness in solving robot path planning problems, thereby broadening its practical utility in various real-world scenarios.

The remaining sections are organized as follows: Section 2 focuses on modeling robot path planning problems and introduces the principles of cubic B-Spline curve-based path smoothing. Section 3 presents an overview of the principles and optimization process of the standard WHO. Section 4 provides a detailed description of the enhanced WHO using hybrid strategies, including the implementation steps and time complexity analysis of the improved HI-WHO. In Section 5, the improved algorithm is evaluated and analyzed for its optimization capabilities using 20 different types of benchmark functions. Section 6 presents the experimental results and discussions of the HI-WHO applied in robot path planning. Finally, Section 7 concludes the study and suggests future directions for research.

2 PROBLEM STATEMENT OF MOBILE ROBOT PATH PLANNING

2.1 Grid map environment modeling

This study aims to utilize the proposed HI-WHO for addressing the path planning problem in a two-dimensional environment containing a finite number of static obstacles. The primary objective is to enable the robot to generate an optimal path from its initial position to the designated target while ensuring collision avoidance. Various modeling methods, such as the octree, grid, topology, and free space, are commonly employed for this purpose (Wu, Huang, Cui, Liu, & Xiao, 2023). Among them, the grid method is a widely used and straightforward approach in robot path planning, which offers the advantage of reducing the complexity of the environment model, while still achieving satisfactory results. Hence, the grid-based methodology is employed for modeling the operational environment of the robot, as illustrated in Figure 1. This study focuses on global path planning for mobile robots in such environments and scenarios, which are also applicable to various practical applications such as unmanned workshops, intelligent warehouses, and specialized environment operations (García, Villar, Tan, Sedano, & Chira, 2023). Furthermore, several assumptions are considered during the formulation of the mathematical model for this problem:

1. The operating environment for the robot is a two-dimensional finite space.
2. The robot is treated as a point mass, and its size and shape can be disregarded.
3. The operational space of the robot contains a finite number of static obstacles whose positions are known, and their heights are not considered.

In Figure 1, the black regions indicate the positions of obstacles in the environment, while the white regions represent the robot’s navigable area. The robot can move in eight neighboring directions from its central position. The objective for the robot is to navigate from the starting point indicated by the circle in the lower-left corner to the destination represented by the star in the upper-right corner. The environment map, denoted as $G$, can be represented by an $N \times N$ binary matrix, where 0 denotes the navigable region, and 1 denotes the static obstacle (Hou, Xiong, Wang, & Chen, 2022). Additionally, each grid on the map is sequentially numbered from left to right and top to bottom as \{1, 2, ..., $N \times N$\}. The center coordinates ($x$, $y$) of any grid $n$ can be calculated using Equations (1) and (2).

$$
\begin{align*}
    x &= \begin{cases} 
        n\%N - 0.5, & n\%N \neq 0 \\
        N - 0.5, & n\%N = 0 
    \end{cases} \\
    y &= \begin{cases} 
        N - 0.5 - \left\lfloor \frac{n}{N} \right\rfloor, & n\%N \neq 0 \\
        N + 0.5 - \left\lfloor \frac{n}{N} \right\rfloor, & n\%N = 0 
    \end{cases}
\end{align*}
$$

(1)

(2)

In which $\%$ denotes the modulo operation, and $\lfloor \cdot \rfloor$ represents the floor function.
2.2 Establishment of objective function

The objective function serves to find the optimal solution for the global path planning problem of the robot, as the fitness value quantifies the evaluation of the solution and its performance. When conducting path planning in a two-dimensional grid map environment, the following conditions must be met:

1. The planned path must be confined within the boundaries of the grid map, ensuring that the path does not extend beyond the map edges.
2. The length of the planned path should be minimized, aiming to achieve an optimal path for the robot.
3. The planned path should avoid traversing through areas occupied by obstacles in the map, preventing collisions during the robot’s movement.

Under the given constraints, the robot aims to plan an optimal collision-free path from the starting point $S(x_s, y_s)$ to the target point $E(x_e, y_e)$ within a predefined grid map space. The intermediate points of the generated path can be denoted as $P_i \in \{(x_i, y_i)\}, i = 1, \ldots, n$. Therefore, the robot’s planned path can be represented as a sequence comprising the starting point, the target point, and the intermediate path points, i.e., $\{S, P_1, P_2, \ldots, P_n, E\}$. Finally, the planned path is obtained by sequentially connecting these points.

Based on the above analysis, the problem of robot path planning in a grid-based environment can be formulated as a single-objective optimization problem, aiming to minimize the fitness value. The objective function for the robot path planning problem, taking into account the aforementioned constraints, is expressed by Equation (3).

$$
\text{Minimize} \quad F = \sum_{i=1}^{n} D_i + O(P_i) \times M.
$$

Subject to

$$
\begin{align*}
x_{\min} & \leq x_i \leq x_{\max} \\
y_{\min} & \leq y_i \leq y_{\max}
\end{align*}
$$

where $D_i$ represents the length of the robot’s planned path at the $i$-th iteration, calculated as the Euclidean distance between the current position and the position at the previous iteration, defined as Equation (4). $O(P_i)$ denotes the penalty value for the $i$-th path planning point. If the point intersects with an obstacle, exhibits a reversal, or exceeds the boundaries, it is considered an invalid path point, as depicted in Figure 2. In such cases, the penalty value assigned to the path point is $N \times N$. Otherwise, it is set to 0. The specific definition is given by Equation (5). Here, $M$ represents the number of path points in the planned path that
intersects with obstacles.

\[
D_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}
\]  
\tag{4}

\[
O(P_i) = \begin{cases} 0, & P_i \in allowed_p \\ N \times N, & \text{otherwise} \end{cases}
\]  
\tag{5}

**FIGURE 2** Three kinds of invalid path points.

### 2.3 Path smoothing

In order to enhance the suitability of the path generated by the HI-WHO for robot navigation and motion control, it is necessary to perform path smoothing on the obtained path. The path smoothing techniques employed in this study primarily involve removing sharp curves from the path and applying cubic B-Spline curve interpolation to the refined path.

First, the slope between a path point \( P_i \) and its adjacent points \( P_{i-1} \) and \( P_{i+1} \) is computed. If the path between these three points is not horizontal or vertical, it indicates the presence of a turning point that requires smoothing. Next, the coordinates of the adjacent points to the turning point are calculated, and their existence in the grid-based map \( G \) is checked for the presence of obstacles. The turning point is considered redundant and can be removed if no obstacles are found. This process is repeated until all redundant turning points in the path are eliminated.

Finally, the processed path is further smoothed using B-Spline curves, which possess advantages such as geometric invariance, convex hull property, convexity preservation, and local support (Wang & Guo, 2023; J. Zhao & Luo, 2022). The expression of the B-Spline curve is given by Equation (6).

\[
C(x) = \sum_{i=1}^{n} Q_i B^k_i(x)
\]  
\tag{6}

in which, \( Q_i \) represents the coordinates of the \( i \)-th control point, and \( B^k_i(x) \) denotes the basis function of a \( k \)-th-order B-Spline curve. A larger value of \( B^k_i(x) \) indicates a smoother curve but requires more computational effort. In this study, a cubic B-Spline curve is employed, which corresponds to \( k = 3 \). The basis function for a cubic B-Spline curve is as follows:

\[
B^3_i(x) = \frac{1}{k!} \sum_{i=1}^{k-i} (-1)^j C^j_{k+1}(x + k - i - j)^k
\]  
\tag{7}
The set of $n$ path points generated by the HI-WHO is denoted as $P_i$ (where $i = 1, 2, ..., n$). For each segment of $k + 1$ adjacent positions, the corresponding B-Spline curve can be calculated through a linear combination, as shown in Equation (8).

$$C_i(x) = \sum_{j=0}^{k} B^k_j(x) Q_{ij}$$  \hspace{1cm} (8)

where $C_i(x)$ represents the $i$-th B-Spline curve. When $k = 3$, the matrix form of the curve can be obtained by Equation (9).

$$C_i(x) = \frac{1}{6} \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{bmatrix}$$  \hspace{1cm} (9)

To account for cases where the path has fewer than four control points at the beginning or end, the endpoint constraint conditions, as shown in Equation (10), are introduced to enable the application of cubic B-Spline interpolation.

$$\begin{cases} P_0 = P_1 \\ P_n = P_{n-1} \end{cases}$$  \hspace{1cm} (10)

Therefore, the coordinates of any point on the cubic B-Spline curve can be calculated using Equation (11).

$$\begin{cases} X_i = \sum_{j=1}^{n} X_j B^3_j(t) \\ Y_i = \sum_{j=1}^{n} Y_j B^3_j(t) \end{cases}$$  \hspace{1cm} (11)

where $(X_i, Y_i)$ represents the coordinates of the cubic B-Spline curve obtained as $t$ varies from 0 to 1. $(X_j, Y_j)$ represents the coordinates of the control points, and $n$ denotes the number of control points.

The smoothing effect of using cubic B-Spline curve is illustrated in Figure 3.

![Figure 3](image.png)

**FIGURE 3** Smoothing effect of using cubic B-Spline curve.

### 3 | STANDARD WILD HORSE OPTIMIZER

The Wild Horse Optimizer (WHO), proposed by Naruei et al. in 2021 (Naruei & Keynia, 2022), is a novel intelligent optimization algorithm that simulates the life behavior of a wild horse population in non-territorial locations for optimization purposes. Typically, a horse herd consists of a stallion, a mare, and foals. The stallion tends to stay near the mare for mating, while the foals begin grazing after birth until they grow up and leave the herd. Male foals join a bachelor herd until they mature enough for mating, while female foals leave the herd and join other herds. This behavior prevents mating between the stallion and female foals or offspring with the same parents. The leader in the population, usually the mare, determines the movement speed and direction of the herd in non-territorial areas. The algorithm exhibits distinct characteristics in the following five aspects.
3.1 Population initialization and group formation with leader selection

Similar to other swarm intelligence optimization algorithms, the WHO initializes the population by generating a random array of individuals within the search space, denoted as \( X = X_1, X_2, ..., X_n \). The target solutions are represented by \( O = O_1, O_2, ..., O_n \), where \( N \) denotes the population size.

The initial population is divided into several groups using Equation (12), and the number of leaders \( N_{Stallion} \) is determined by the number of groups formed by the stallion population, denoted as \( G \). The remaining horse groups in the original population, including mares and foals, are evenly distributed with \( N_{Foal} = N - G \) individuals.

\[
G = \left\lfloor N \times PS \right\rfloor
\]  

where the control parameter \( PS \) represents the percentage of the stallion population partitioned from the initial population. Initially, leaders are randomly selected. During the algorithm execution, leaders are chosen based on the fitness values of the horse group members.

3.2 Grazing behavior

The foals typically spend most of their time grazing around a group of horses. The algorithm simulates the grazing behavior of wild horses using Equation (13), where the position of the stallion is considered the center of the grazing area, and the other group members (mares and foals) graze around the center. They move and search around the leader within different radii.

\[
X_{i,G}^j = 2Z \cos(2\pi RX) \times (X_{Stallion}^j - X_{i,G}^j) + X_{Stallion}^j
\]  

where \( X_{Stallion}^j \) represents the position of the stallion (leader) within the group, \( X_{i,G}^j \) represents the current position of the group member (mare or foal), \( Z \) is an adaptive parameter calculated using Equation (14), and \( R \) is a uniform random number ranging from -2 to 2. This parameter allows the group to graze around the leader within a 360-degree radius. The individuals in the population can move within different radii by using the \( \cos \) function. \( X_{i,G}^j \) denotes the updated position of the group member after the grazing behavior.

\[
\begin{align*}
P &= \bar{R}_1 < TDR; \\
IDX &= (P == 0); \\
Z &= R_2IDX + \bar{R}_3\Theta(\sim IDX).
\end{align*}
\]  

where \( P \) is a vector containing zeros and ones, \( \bar{R}_1 \) and \( \bar{R}_3 \) are random vectors with uniform distribution. The values ranging from 0 to 1, \( R_2 \) is a uniform random number ranging from 0 to 1, \( IDX \) represents the index values of elements in vector \( \bar{R}_1 \) that satisfy \( (P == 0) \), and \( TDR \) is an adaptive parameter. As the number of iterations increases, the value of \( TDR \) decreases from 1 to 0, which is calculated as:

\[
TDR = 1 - \frac{t}{T_{max}}
\]  

where \( t \) represents the current iteration number, and \( T_{max} \) is the maximum number of iterations.

3.3 Mating behavior

Once the male and female foals from different groups mature, they can mate and reproduce offspring. The offspring produced need to leave their current group and join another group. The algorithm simulates the mating behavior of wild horses using the mean crossover operator, as shown in Equation (16).

\[
\begin{align*}
X_{G,k}^o = & \text{Crossover}(X_{G,i}^p, X_{G,j}^q); \\
i \neq j \neq k, o = p = \text{end}; \\
\text{Crossover} = \text{Mean}.
\end{align*}
\]
where $X_{G,k}^o$ represents the position of individual $o$ in group $k$. It leaves group $k$ and is replaced by offspring generated through mating between individuals $p$ in group $i$ and $q$ in group $j$. $X_{G,i}^p$ denotes the position of foal $p$ in group $i$, and $X_{G,j}^q$ denotes the position of foal $q$ in group $j$. These foals mature and leave their respective groups to engage in mating.

### 3.4 Group leadership

Group leaders lead their groups toward more suitable locations (watering hole positions). If the current group occupies a dominant position, they continue to use that area. However, if another group occupies the area, the leader needs to guide the current group away from that area. This process is expressed in Equations (17) and (18).

\[
\bar{X}_{Stallion}^i = 2Z \cos(2\pi RZ) \times (WH - X_{Stallion}^i) + WH, R_5 > 0.5
\]

\[
\bar{X}_{Stallion}^i = Z \cos(2\pi RZ) \times (WH - X_{Stallion}^i) - WH, R_5 \leq 0.5
\]

where $\bar{X}_{Stallion}^i$ represents the next position of the leader in group $i$, $WH$ denotes the position of the most suitable area (watering hole), $X_{Stallion}^i$ represents the current position of the leader in group $i$, $Z$ is an adaptive parameter calculated using Equation (14), $R$ is a random number between -2 and 2, and $\pi$ equals 3.14.

### 3.5 Leader exchange and selection

The group leaders are initially randomly selected and are then updated based on the fitness values during the iterations of the algorithm. If the fitness of a group member is superior to that of the leader, the positions of the leader and the corresponding group member are updated according to Equation (19).

\[
X_{Stallion}^i = \begin{cases} 
\bar{X}_{Stallion}^i, & \text{cost}(\bar{X}_{Stallion}^i) < \text{cost}(X_{Stallion}^i) \\
X_{Stallion}^i, & \text{cost}(\bar{X}_{Stallion}^i) \geq \text{cost}(X_{Stallion}^i) 
\end{cases}
\]

### 4 HYBRID STRATEGY IMPROVED WILD HORSE OPTIMIZER (HI-WHO)

#### 4.1 Sobol sequence to initialize populations

The initial population distribution is crucial for the search efficiency and optimization accuracy of metaheuristic algorithms. Generally, a more balanced initial population distribution leads to higher search efficiency. The standard WHO uses a random approach to generate the initial population with low coverage and uneven distribution. In this paper, the population is initialized using low-discrepancy sequences, guiding the foal population to be orderly distributed around the leader horse. Common methods for generating such sequences include Sobol, Halton, and Faure sequences. Among them, Halton and Faure methods exhibit a high correlation between sequences when generating high-dimensional sequences and require relatively longer computation time. Therefore, in this study, the Sobol sequence is selected for initializing the wild horse population due to its advantages of uniform population distribution, efficient computation, and broader sampling range, as shown in Equation (20).

\[
X_n = LB + S_n \cdot (UB - LB)
\]

where $n$ is an integer ranging from 1 to $N$, $[LB, UB]$ represents the range of the solution, $S_n$ is a Sobol sequence ranging from 0 to 1, and $X_n$ represents the initial population generated using the Sobol sequence.

Suppose the population size is 100, and the lower and upper bounds of the solution range are 0 and 1, respectively. The comparison between the initialization of the population using pseudorandom numbers and the Sobol sequence is shown in Fig. 4. It can be observed from Figure 4 that the population initialized using the Sobol sequence exhibits a more uniform distribution and covers a wider range compared to the population initialized using pseudorandom numbers.
4.2  |  Lévy flight strategy for individual position update

The movement of individuals (foals and mares) in the wild horse optimization algorithm involves updating their positions, which is primarily influenced by the position of the group leader (stallion). This phase focuses on manipulating individuals with lower fitness values in the population. However, the original algorithm’s position update method is limited, which can lead to algorithm stagnation. The Lévy flight strategy is introduced to update the positions of the horse herd individuals, drawing inspiration from the Cuckoo Search (CS) algorithm [Gandomi, Yang, & Alavi, 2013]. When a random value $R_4$ is greater than the mating probability $PC$, the population individuals undergo position updates through the Lévy flight strategy, as described by Equation (21). When $R_4$ is less than $PC$, the horse mating behavior position update method is used as shown in Equation (16).

$$X_{i,G}^j = \alpha (X_{Stallion}^j - Xa_{i,G}^j) \oplus \text{Levy}(\delta) + X_{Stallion}^j, \quad R_4 > PC$$

(21)

where $R_4$ is a random value between 0 and 1, $\alpha$ is the step size adjustment coefficient with a value of 0.01, $\oplus$ denotes element-wise multiplication, and $\text{Levy}(\delta)$ represents a path that follows the Lévy distribution. The Lévy distribution is defined as:

$$\text{Levy}(\delta) = \frac{\mu}{|\nu|^{\frac{1}{\delta}}}$$

(22)

where $\mu$ and $\nu$ follow normal distribution as described in Equations (23) and (24):

$$\mu \sim N(0, \sigma_{\mu}^2), \nu \sim N(0, \sigma_{\nu}^2)$$

(23)

$$\sigma_{\mu} = \left[ \frac{\Gamma(1 + \delta) \sin(\delta \pi/2)}{\Gamma((1 + \delta)/2) \delta^{1/2}} \right]^{1/\delta}, \sigma_{\mu} = 1$$

(24)

where $\Gamma$ represents the gamma function, and $\delta$ is a value ranging from 0 to 2. In this paper, we choose the value of $\delta = 1.5$.

4.3  |  Nonlinear dynamic self-adaptive factor

From Equation (15), it can be inferred that a linear adaptive factor is employed to control the variable $Z$ in the standard WHO, which is crucial for balancing global exploration and local exploitation. This study introduces a non-linear dynamic adaptive factor, as presented in Equation (25), that adapts its magnitude with increased algorithm iterations. Initially, the adaptive factor is set to a relatively large value, ensuring a rapid descent rate and efficient global search within the search space. As the algorithm progresses, the adaptive factor gradually decreases, enhancing the algorithm’s ability to perform detailed local searches and effectively resolving the problem of getting trapped in local optima and stagnation during the later iterations.

$$TDR = [1 + \cos(\frac{\pi}{2} \cdot \frac{t}{T_{max}} + \frac{\pi}{2})]^{dnw}$$

(25)
where \( t \) represents the current iteration number of the algorithm, \( T_{\text{max}} \) is the maximum number of iterations, and \( dnw \) denotes the dynamic non-linear adjustment factor.

The relationship between \( dnw \) and the variation of TDR is shown in Figure 5 when \( dnw \) takes different values. The graph illustrates the change in TDR for \( dnw \) values ranging from 0.2 to 3.4. It can be observed that different values of \( dnw \) result in significant variations in the curve of TDR, all exhibiting nonlinear patterns. As the value of \( dnw \) increases, the curve becomes more concave, leading to a rapid initial decline of TDR and a stronger global search capability of the algorithm with higher search efficiency. In the later iterations, the change in TDR becomes smoother, enabling the algorithm to perform fine-grained local search and improve its precision. Through repeated testing, a value of 2.0 for \( dnw \) achieves a good balance between global exploration and local exploitation in the algorithm.

![Figure 5: TDR curves with different \( dnw \) values.](image)

### 4.4 Lens imaging opposition-based learning strategy

The WHO initially disperses the horse population individuals within the search space, exhibiting strong global exploration capability. However, as the algorithm iterates, Equations (17) and (18) indicate that the horse population individuals gradually converge towards the optimal position under the influence of the group leadership. Consequently, the horse population becomes concentrated in a smaller area, reducing in population diversity. If the group leader’s position corresponds to a local optimal solution at this stage, the algorithm may prematurely converge and get stuck in a local optimum. This paper introduces the lens imaging opposition-based learning strategy (Bo, Cheng, & Khishe, 2023), which combines the individuals generated through reverse action with the current optimal individual of the algorithm. The specific process is as follows:

Suppose that an individual \( P \) exists in the range \([LB, UB]\) with a height of \( h \), and its projection on the coordinate axis is denoted as \( X \) (representing the global optimal individual). Next, a convex lens with a focal length of \( f \) is placed in the one-dimensional space at the base point \( O \) (the midpoint of LB and UB). After being refracted by the convex lens, the individual \( P \) generates an image \( P' \) with a height of \( h' \), and its projection on the coordinate axis is denoted as \( X' \). This image \( P' \) represents the opposite individual obtained through the lens imaging opposition-based learning strategy from the global optimal individual \( X \), as shown in Figure 6.

The above figure shows the relationship between the global optimum individual \( X \) and its corresponding reverse individual \( X' \), which can be obtained through the principle of convex lens imaging as Equation (26).

\[
\frac{(LB + UB)/2 - X}{X' - (LB + UB)/2} = \frac{h}{h'}
\]  

(26)
Figure 6 Illustration of lens imaging opposition-based learning strategy.

where \( \lambda = \frac{h}{h'} \) is the scaling factor between the heights of the original individual \( P \) and the image individual \( P' \). By performing the necessary transformations on Equation (26), the calculation formula for the reverse individual \( X' \) can be obtained as follows.

\[
X' = \frac{LB + UB}{2} + \frac{LB + UB}{2} \cdot \frac{X}{\lambda}
\]  

(27)

When \( \lambda = 1 \), the above equation transforms into the standard opposition-based learning formula, which is:

\[
X' = LB + UB - X
\]  

(28)

According to Equation (28), it represents a specific case of the lens imaging opposition-based learning strategy combined with \( X \) within the range of \([\frac{-LB+UB}{2}, \frac{LB+UB}{2}]\), where the obtained candidate solution is relatively fixed. In order to increase the population diversity of the algorithm and enhance its search range and local development capability, this study adopts a dynamic nonlinear scaling factor as shown in Equation (29), which dynamically adjusts \( \lambda \) to obtain candidate solutions that are updated dynamically.

\[
\lambda = \lambda_{\text{min}} + (\lambda_{\text{max}} - \lambda_{\text{min}}) \cdot (1 - \frac{t}{T_{\text{max}}})^2
\]  

(29)

where \( \lambda_{\text{min}} \) and \( \lambda_{\text{max}} \) represent the minimum and maximum scaling factors of the lens imaging opposition-based learning strategy, respectively, and \( T_{\text{max}} \) denotes the maximum number of iterations for the algorithm.

If the search space is extended to high dimensions, the following equation can be obtained:

\[
X'_{ij} = \frac{LB_j + UB_j}{2} + \frac{LB_j + UB_j}{2} \cdot \frac{X_{ij}}{\lambda}
\]  

(30)

where \( X_{ij} \) represents the position of individual \( i \) in the population on dimension \( j \), \( X'_{ij} \) represents the reverse solution obtained by applying the lens imaging opposition-based learning strategy to \( X_{ij} \), and \( LB_j \) and \( UB_j \) represent the lower and upper bounds of the decision variable on dimension \( j \).

Although applying the lens imaging opposition-based learning strategy to the standard WHO can expand the search range and improve the ability to escape local optima, it is difficult to determine whether the generated opposite individuals are better than the current best. Therefore, this paper uses a greedy mechanism to further select the better individual between them based on the fitness value. By continuously updating the best solution, the optimization ability of the algorithm is improved. This process can be represented by Equation (31).

\[
X_{\text{new}}(t) = \begin{cases} 
X, & \text{cost}(X) < \text{cost}(X') \\
X', & \text{cost}(X) \geq \text{cost}(X')
\end{cases}
\]  

(31)

4.5 Implementation and flowchart of the HI-WHO

The pseudocode of the HI-WHO, incorporating the aforementioned improved strategies, is presented in Algorithm 1. Moreover, the flowchart is depicted in Figure 7.
Algorithm 1 Pseudo-code of hybrid strategy improved Wild Horse Optimizer

\textbf{Input:} Parameters for HI-WHO: population size \(N\), search space dimension \(d\) and range \([LB, UB]\), maximum iterations \(T_{\text{max}}\).

\textbf{Output:} The best optimal value that is the position of the best horse.

1: Initialize the leader ratio parameter \(PS\), the mating probability parameter \(PC\), \(N_{\text{Stallion}}\) and \(N_{\text{Foal}}\).
2: Create foal groups with the Sobol sequence and select stallions
3: Calculate the fitness of each horse
4: \textbf{procedure} HI-WHO
5: \hspace{1em} while \(t < T_{\text{max}}\) do
6: \hspace{2em} Calculate the value of \(TDR\) by (25)
7: \hspace{2em} for \(i = 1\) to \(N_{\text{Stallion}}\) do
8: \hspace{3em} if \(R_4 > PC\) then
9: \hspace{4em} Update the position of the foal using Lévy flight strategy by (21)
10: \hspace{3em} else
11: \hspace{4em} Update the position of the foal with mean crossover operator by (16)
12: \hspace{3em} end if
13: \hspace{2em} end for
14: \hspace{2em} if \(R_5 > 0.5\) then
15: \hspace{3em} Update the position of the stallion \(X_{\text{Stallion}}^i\) by (17)
16: \hspace{2em} else
17: \hspace{3em} Update the position of the stallion \(X_{\text{Stallion}}^i\) by (18)
18: \hspace{2em} end if
19: \hspace{2em} if \(\text{cost}(X_{\text{Stallion}}^i) < \text{cost}(X_{\text{Stallion}})\) then
20: \hspace{3em} \(X_{\text{Stallion}} = X_{\text{Stallion}}^i\)
21: \hspace{2em} end if
22: \hspace{2em} Calculate the fitness and sort the foals by \text{cost}
23: \hspace{2em} Select the foal with minimum \text{cost}
24: \hspace{2em} Update the best foal with Lens imaging opposition-based learning strategy by (30)
25: \hspace{2em} if \(\text{cost}(X_{\text{Foal}}^i) < \text{cost}(X_{\text{Foal}})\) then
26: \hspace{3em} \(X_{\text{Foal}} = X_{\text{Foal}}^i\)
27: \hspace{2em} end if
28: \hspace{2em} if \(\text{cost}(X_{\text{Foal}}) < \text{cost}(X_{\text{Stallion}})\) then
29: \hspace{3em} Exchange the position of foal and stallion by (19)
30: \hspace{2em} end if
31: \hspace{2em} end while
32: \hspace{2em} Update and output the global optimum
33: \textbf{end procedure}

4.6 Complexity analysis

Time complexity is an important metric for assessing and evaluating the efficiency of an algorithm, and it indirectly reflects the convergence speed of the algorithm. Space complexity refers to the amount of storage space required by an algorithm during its execution. The space complexity helps assess the efficiency and scalability of an algorithm in terms of memory usage. In the standard WHO, assuming the total number of iterations is \(T_{\text{max}}\), the population size is \(N\), and the dimension of individuals is \(n\).
4.6.1 Time complexity analysis of WHO

In the initialization stage of the WHO, the time required for initializing basic parameters is denoted as $\eta_0$. Additionally, generating uniformly distributed random numbers for each dimension of individual positions takes $\eta_1$, calculating the fitness value for each individual based on the objective function takes $f(n)$, and the computation time for grouping the horse population and selecting leaders is denoted as $\eta_2$. The overall time complexity of this stage can be expressed as $T_1 = O(\eta_0 + N(n\eta_1 + f(n)) + \eta_2) = O(n + f(n))$.

During the grazing behavior stage of the horses, the population size of foals is $0.8N$. The time required to calculate the $TDR$ value is $\eta_3$, while the time for calculating the $Z$ value is $\eta_4$. Additionally, updating the foal positions in each dimension takes $\eta_5$, as indicated by Equation (13). Therefore, the time complexity of this stage can be expressed as $T_2 = O(\eta_3 + 0.8N(n\eta_4 + \eta_5)) = O(n)$. While in the mating behavior stage of the horses, the foal positions are updated using Equation (16), which requires a computation time of $\eta_6$. The time required to calculate the fitness value of an individual remains as $f(n)$. Therefore, the time complexity of this stage can be expressed as $T_3 = O(0.8N(n\eta_6 + f(n))) = O(n + f(n))$.

During the leader exchange and selection stage, the population size of stallions is $0.2N$. The time required to update the leader positions using Equations (17) and (18) is $\eta_7$, the time required to calculate the fitness value of individuals is $f(n)$,
and the time required for leader exchange and selection using Equation (19) is \( t_{8} \). The time complexity of this stage is 
\[
T_{2} = O(0.2N(nT_{1} + f(n)) + nT_{8}) = O(n + f(n)).
\]

Therefore, the total time complexity of the standard WHO is 
\[
T(n) = T_{1} + T_{max}(T_{2} + T_{3} + T_{4}) = O(n + f(n)).
\]

### 4.6.2 Time complexity analysis of HI-WHO

In the proposed HI-WHO, the population size, individual dimensions, time for initializing basic parameters, time for generating a uniform random number, and time for calculating the fitness value of the objective function are all the same as in the standard WHO. During the initialization stage, the time required to initialize the foal population using the Sobol sequence according to Equation (20) is denoted as \( t_{0} \). The time complexity of this stage is 
\[
T_{1}' = O(\eta_{0} + 0.2Nm_{T_{1}} + 0.8Nnt_{0} + Nf(n) + \eta_{2}) = O(n + f(n)).
\]

In the grazing behavior stage of the HI-WHO, the TDR value is computed using Equation (25), with the calculation time denoted as \( t_{1} \). The calculation time for updating the positions of foals using the Lévy flight strategy, as per Equation (21), is denoted as \( t_{3} \). Hence, the time complexity of this stage is 
\[
T_{2}' = O(t_{1} + 0.8NnT_{1} + t_{2}) = O(n).
\]

During the mating behavior stage, the time complexity remains the same as the original WHO, i.e., \( T_{3} = T_{3} = O(n + f(n)) \).

Lastly, in the leader exchange and selection stage, the time complexity of updating the optimal foal positions using the lens reverse learning strategy, as described in Equation (30), is denoted as \( t_{4} \). The time complexity of performing the greedy mechanism based on Equation (31) is denoted as \( t_{4}' \). The remaining computation time remains unchanged. Therefore, the time complexity of this stage is 
\[
T_{4}' = O(0.2N(nT_{1} + f(n)) + n(t_{3} + t_{4}) + nT_{8}) = O(n + f(n)).
\]

Thus, the total time complexity of the HI-WHO is given by 
\[
T(n) = T_{1}' + T_{max}(T_{2}' + T_{3}' + T_{4}') = O(n + f(n)).
\]

Compared to the standard WHO, the proposed HI-WHO has a slightly increased execution time but maintains the same time complexity. The mechanism improvements in HI-WHO do not reduce the algorithm’s running efficiency.

### 4.6.3 Space complexity analysis

For metaheuristic algorithms, the space complexity is determined by population size and problem dimensions. In the standard WHO, the population size is \( N \) and the problem dimension is \( n \), the complexity is \( O(Nn) \). Similarly, the HI-WHO still uses a population of size \( N \), and the problem dimension remains \( n \), so the space complexity of HI-WHO is also \( O(Nn) \). Therefore, regarding space complexity, the HI-WHO is the same as WHO. The HI-WHO does not require additional storage space to enhance its performance.

### 5 BENCHMARK FUNCTION TESTING AND RESULT ANALYSIS

#### 5.1 Benchmark functions

This paper selected 15 classic benchmark functions from [Shen, Zhang, Gharehchopogh, & Mirjalili, 2023], [S. Zhao, Zhang, Ma, & Wang, 2023], [Hu, Wang, Li, Yang, & Zheng, 2023] and 5 CEC2022 benchmark functions from [Abdel-Basset, Mohamed, Sallam, & Chakrabortty, 2022] to assess the effectiveness of the improved HI-WHO in addressing global optimization problems. Based on the chosen classic benchmark function features, they were classified into three categories. Functions \( F_{1} \) – \( F_{5} \) belong to the unimodal functions, characterized by having a single global optimum value. Consequently, these functions enable the assessment of the exploitation capacity and convergence speed of the algorithm. Functions \( F_{6} \) – \( F_{10} \) represent the multimodal functions, which possess multiple local optimum values, making the identification of the global optimum challenging. These functions are utilized to evaluate the global exploration ability. Functions \( F_{11} \) – \( F_{15} \) are fixed-dimension multipeak functions featuring multiple extremal points. However, due to their low dimensions, optimization is relatively easier, allowing for the assessment of algorithm stability and the ability to explore in low dimensions. Among the selected 5 CEC2022 benchmark functions, they include a unimodal function \( F_{16} \), two basic functions \( F_{17} – F_{18} \), a hybrid function \( F_{19} \), and a composite function \( F_{20} \). The function names, expressions, dimensions, search ranges, and theoretical optimal values for each benchmark function are provided in Tables 1 to 4.
5.2 | Experimental parameter settings

In order to evaluate the performance of the improved HI-WHO, comparative testing was conducted on different benchmark functions using six algorithms: the standard Wild Horse Optimizer (WHO) [Naruei & Keynia, 2022], the improved WHO (IWHO) [Zheng et al., 2022], the Lévy flight-based improved WHO (IWHOLF) [Saravanan, Neelakandan, Ezhumalai, & Maurya, 2023], the improved Particle Swarm Optimization (IPSO) algorithm [Song, Wang, & Zou, 2021], and the improved Sparrow Search Algorithm (ISSA) [G. Zhang & Zhang, 2022]. The search space dimensions for functions $F_1$ – $F_{10}$ were set at 30, 50, and 100 dimensions to compare the optimization performance of the HI-WHO across different search dimensions. For the CEC2022 test suites, the search dimension is set to 10. IWHO and IWHOLF are representative improvements of the original

### Table 1
Unimodal benchmark functions

<table>
<thead>
<tr>
<th>No.</th>
<th>Function name</th>
<th>Expression</th>
<th>Dimension</th>
<th>Range</th>
<th>Optima</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_1$</td>
<td>Sphere</td>
<td>$f_1(x) = \sum_{i=1}^{n} x_i^2$</td>
<td>$30/50/100$</td>
<td>$[-100,100]$</td>
<td>0</td>
</tr>
<tr>
<td>$F_2$</td>
<td>Schwefel 2.22</td>
<td>$f_2(x) = \sum_{i=1}^{n}</td>
<td>x_i</td>
<td>+ \prod_{i=1}^{n}</td>
<td>x_i</td>
</tr>
<tr>
<td>$F_3$</td>
<td>Rosenbrock</td>
<td>$f_3(x) = \sum_{i=1}^{n} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$</td>
<td>$30/50/100$</td>
<td>$[-100,100]$</td>
<td>0</td>
</tr>
<tr>
<td>$F_4$</td>
<td>Step</td>
<td>$f_4(x) = \sum_{i=1}^{n} (x_i + 0.5)^2$</td>
<td>$30/50/100$</td>
<td>$[-100,100]$</td>
<td>0</td>
</tr>
<tr>
<td>$F_5$</td>
<td>Quartic</td>
<td>$f_5(x) = \sum_{i=1}^{n} x_i^4 + \text{random}(0,1)$</td>
<td>$30/50/100$</td>
<td>$[-1.28,1.28]$</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 2
Multimodal benchmark functions

<table>
<thead>
<tr>
<th>No.</th>
<th>Function name</th>
<th>Expression</th>
<th>Dimension</th>
<th>Range</th>
<th>Optima</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_6$</td>
<td>Schwefel 2.26</td>
<td>$f_6(x) = -\sum_{i=1}^{n} x_i \sin(\sqrt{</td>
<td>x_i</td>
<td>})$</td>
<td>$30/50/100$</td>
</tr>
<tr>
<td>$F_7$</td>
<td>Rastrigin</td>
<td>$f_7(x) = \sum_{i=1}^{n} (x_i^2 - 10 \cos(2\pi x_i) + 10)$</td>
<td>$30/50/100$</td>
<td>$[-5.12,5.12]$</td>
<td>0</td>
</tr>
<tr>
<td>$F_8$</td>
<td>Ackley</td>
<td>$f_8(x) = -20 \exp(-0.2 \sum_{i=1}^{n} \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}) - \exp(1 \sum_{i=1}^{n} \cos(2\pi x_i)) + 20 + e$</td>
<td>$30/50/100$</td>
<td>$[-32.07,32.07]$</td>
<td>0</td>
</tr>
<tr>
<td>$F_9$</td>
<td>Griewank</td>
<td>$f_9(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} \cos(x_i) + 1$</td>
<td>$30/50/100$</td>
<td>$[-600,600]$</td>
<td>0</td>
</tr>
<tr>
<td>$F_{10}$</td>
<td>Penalized 1.1</td>
<td>$y_i = 1 + \frac{1}{\alpha_i} (\alpha_i - a)^2$</td>
<td>$30/50/100$</td>
<td>$[-50,50]$</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 3
Fixed-dimension multipeak benchmark functions

<table>
<thead>
<tr>
<th>No.</th>
<th>Function name</th>
<th>Expression</th>
<th>Dimension</th>
<th>Range</th>
<th>Optima</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{11}$</td>
<td>Foxholes</td>
<td>$f_{11}(x) = \left(\frac{1}{\prod_{i=1}^{n} x_i} + \sum_{i=1}^{25} \frac{1}{n - \sum_{j=1}^{n} (x_j - a_j)^2}\right)^{1/2}$</td>
<td>2</td>
<td>$[-65.536,65.536]$</td>
<td>1</td>
</tr>
<tr>
<td>$F_{12}$</td>
<td>Kowalik</td>
<td>$f_{12}(x) = \sum_{i=1}^{10} (x_i^2 + 0.0001) \frac{1}{1 + \prod_{i=1}^{10} (x_i - a_i)^2}$</td>
<td>4</td>
<td>$[-5.5]$</td>
<td>0.0003</td>
</tr>
<tr>
<td>$F_{13}$</td>
<td>Branin</td>
<td>$f_{13}(x) = \left(\frac{x_2 - \frac{1}{\pi} \sin(x_1) + 1}{\frac{18.0316}{\pi}}\right)^{2} + 10 \left(1 - \frac{1}{\pi}\right) \cos(x_1 + 5.5)$</td>
<td>2</td>
<td>$[-5.10,0.15]$</td>
<td>0.3983</td>
</tr>
<tr>
<td>$F_{14}$</td>
<td>Goldstein Price</td>
<td>$f_{14}(x) = \left(1 + (x_1 + x_2)^2 [19 - 14x_1 + 3x_1^2]\right) \times \left(1 + (2x_1 - 3x_2)^2 [18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2]\right)$</td>
<td>2</td>
<td>$[-2.2]$</td>
<td>3</td>
</tr>
<tr>
<td>$F_{15}$</td>
<td>Shekel10</td>
<td>$f_{15}(x) = -\sum_{i=1}^{10} (X_i - a_i)(X_i - a_i)^T + c_i$</td>
<td>4</td>
<td>$[0,10]$</td>
<td>-10.5363</td>
</tr>
</tbody>
</table>

### Table 4
CEC2022 benchmark functions

<table>
<thead>
<tr>
<th>No.</th>
<th>Function name</th>
<th>Class</th>
<th>Range</th>
<th>Optima</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{16}$</td>
<td>Shifted and full rotated Zakharov function</td>
<td>Unimodal</td>
<td>$[-100,100]$</td>
<td>300</td>
</tr>
<tr>
<td>$F_{17}$</td>
<td>Shifted and full rotated expanded Scaffer’s F6 function</td>
<td>Multimodal</td>
<td>$[-100,100]$</td>
<td>600</td>
</tr>
<tr>
<td>$F_{18}$</td>
<td>Shifted and full rotated non-continuous Rastrigin’s function</td>
<td>Multimodal</td>
<td>$[-100,100]$</td>
<td>800</td>
</tr>
<tr>
<td>$F_{19}$</td>
<td>Hybrid function 3 (N = 5)</td>
<td>Hybrid</td>
<td>$[-100,100]$</td>
<td>2200</td>
</tr>
<tr>
<td>$F_{20}$</td>
<td>Composite function 4 (N = 6)</td>
<td>Composite</td>
<td>$[-100,100]$</td>
<td>2700</td>
</tr>
</tbody>
</table>
WHO, while IPSO and ISSA are recent and effective metaheuristic algorithms used for robot path planning. All six algorithms are independently executed on the same software and hardware environment. The experimental hardware platform consisted of a host machine with the Windows 10 64-bit operating system, an Intel(R) Core(TM) i7-8550U CPU @1.8GHz, and 32GB of memory. The software platform used is Matlab R2020a.

Regarding the algorithm parameters, the population size was set to \( N = 50 \), and the maximum number of iterations was set to \( T_{\text{max}} = 1000 \) for all algorithms. Other specific parameters for each algorithm are kept consistent with the original literature, as shown in Table 5.

### TABLE 5 Parameter settings

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHO</td>
<td>( PC = 0.13 ), ( PS = 0.2 )</td>
</tr>
<tr>
<td>IWHO</td>
<td>( PC = 0.13 ), ( PS = 0.2 ), ( PRR = 0.1 ), ( \omega \in [0.01, 0.99] )</td>
</tr>
<tr>
<td>IWHOLF</td>
<td>( PC = 0.13 ), ( PS = 0.2 )</td>
</tr>
<tr>
<td>HI-WHO</td>
<td>( PC = 0.13 ), ( PS = 0.2 ), ( dw = 2.0 )</td>
</tr>
<tr>
<td>IPSO</td>
<td>( c1 = c2 = 2 ), ( w = 1.2 )</td>
</tr>
<tr>
<td>ISSA</td>
<td>( ST = 0.8 ), ( PD = 0.3 ), ( SD = 0.2 )</td>
</tr>
</tbody>
</table>

The experimental data were obtained by running each algorithm independently 50 times and then collecting the statistical results. The evaluation metrics employed in the testing included the minimum value (\( \text{Min} \)), the average value of optimization accuracy (\( \text{Mean} \)), and the standard deviation (\( \text{Std} \)). The minimum value reflects the optimization capability and feasibility of the algorithm, while the mean value represents the algorithm’s accuracy. The standard deviation reflects the stability of the algorithm.

The calculation formulas for the mean value and standard deviation are given in Equation (32) and Equation (33), respectively.

\[
\text{Mean} = \frac{1}{n} \sum_{i=1}^{n} F_i
\]  
(32)

\[
\text{Std} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (F_i - \text{Mean})^2}
\]  
(33)

### 5.3 Analysis of experimental results and comparison of algorithm performance

#### 5.3.1 Analysis of optimization accuracy

To test and evaluate the optimization performance of the HI-WHO, the dimensions of functions \( F_1 \) to \( F_{10} \) are set to \( D = 30 \), 50, and 100 for the extremal optimization experiments. Each algorithm is independently executed 50 times under the same conditions to ensure the objectivity and sufficiency of the tests. The population size \( N \) is set to 50, and the maximum evolution generations \( T_{\text{max}} \) is set to 1000. Tables 6 and 7 present the minimum value, average value, and variance of the optimization results obtained from running each algorithm 50 times in 30-dimensional, 50-dimensional, and 100-dimensional settings. The optimal values for each item are bolded for clarity.

The experimental results and statistical data from Tables 6 and 7 reveal that the HI-WHO demonstrates superior performance compared to the other five algorithms in handling both uni-modal and multi-modal problems. Specifically, it achieves excellent optimal values in the uni-modal test functions \( F_1 \) and \( F_2 \), as well as the multi-modal test functions \( F_7 \) and \( F_9 \). Except for a few cases in \( F_3 \), \( F_6 \), and \( F_{10} \), the HI-WHO demonstrates significantly superior optimization accuracy and stability compared to the other five algorithms across different dimensions. Moreover, across all dimensions of the ten test functions, the HI-WHO outperforms the standard WHO in terms of the minimum value, average value, and variance of the optimization results from 50 independent runs. These results highlight the notable improvements and effectiveness of the proposed algorithm in enhancing the standard WHO, which suggests that by improving the initial population, the HI-WHO has a broader search space. Furthermore, introducing the Lévy flight strategy enhances local exploitation capabilities and improves global optimization performance. Additionally, the lens imaging opposition-based learning strategy can enhance the algorithm’s ability to escape local optima.
**Table 6** Comparison of optimization results on uni-modal and multi-modal benchmark test functions($F_1 - F_{10}$)

<table>
<thead>
<tr>
<th>Function</th>
<th>D</th>
<th>Min</th>
<th>Mean</th>
<th>Std</th>
<th>D</th>
<th>Min</th>
<th>Mean</th>
<th>Std</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$F_1$</td>
<td>30</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>100</td>
<td>2.242E-141</td>
<td>1.251E-128</td>
<td>6.422E-128</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>100</td>
<td>4.221E-128</td>
<td>2.645E-112</td>
<td>1.867E-111</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>100</td>
<td>7.171E-123</td>
<td>1.280E-107</td>
<td>7.738E-107</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>100</td>
<td>2.138E-77</td>
<td>1.520E-70</td>
<td>4.817E-70</td>
</tr>
<tr>
<td>$F_2$</td>
<td>50</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>100</td>
<td>6.849E-71</td>
<td>6.084E-64</td>
<td>2.264E-63</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>100</td>
<td>4.293E-68</td>
<td>8.324E-61</td>
<td>4.982E-60</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>9.711E-12</td>
<td>1.827E-06</td>
<td>5.057E-06</td>
<td>100</td>
<td>2.300E+01</td>
<td>2.414E+01</td>
<td>3.676E-01</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>7.507E-10</td>
<td>8.227E-05</td>
<td>2.279E-04</td>
<td>100</td>
<td>9.430E+01</td>
<td>9.562E+01</td>
<td>7.276E-01</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>1.154E-21</td>
<td>1.123E-17</td>
<td>2.50E-17</td>
<td>100</td>
<td>3.073E-18</td>
<td>3.799E-14</td>
<td>1.370E-13</td>
</tr>
<tr>
<td>$F_4$</td>
<td>50</td>
<td>8.483E-17</td>
<td>8.523E-12</td>
<td>2.373E-11</td>
<td>100</td>
<td>5.764E-08</td>
<td>2.233E-05</td>
<td>5.495E-05</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>8.072E-13</td>
<td>1.445E-07</td>
<td>8.887E-07</td>
<td>100</td>
<td>5.121E-02</td>
<td>5.126E-01</td>
<td>3.143E-01</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>9.051E-07</td>
<td>3.118E-05</td>
<td>2.960E-05</td>
<td>100</td>
<td>2.144E-05</td>
<td>3.604E-04</td>
<td>2.439E-04</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1.400E-07</td>
<td>3.339E-05</td>
<td>3.569E-05</td>
<td>100</td>
<td>4.118E-05</td>
<td>3.964E-04</td>
<td>2.668E-04</td>
</tr>
<tr>
<td>$F_6$</td>
<td>50</td>
<td>-1.257E+04</td>
<td>-1.200E+04</td>
<td>8.956E-02</td>
<td>100</td>
<td>-1.069E+04</td>
<td>-9.293E+03</td>
<td>5.500E+02</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>-2.095E+04</td>
<td>-1.958E+04</td>
<td>1.905E+03</td>
<td>100</td>
<td>-1.583E+04</td>
<td>-1.420E+04</td>
<td>7.340E-02</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>100</td>
<td>-2.840E+04</td>
<td>-2.485E+04</td>
<td>1.159E+03</td>
</tr>
<tr>
<td>$F_7$</td>
<td>50</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>100</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>100</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>8.882E+16</td>
<td>8.882E+16</td>
<td>0.000E+00</td>
<td>100</td>
<td>8.882E-16</td>
<td>2.522E-15</td>
<td>1.789E-15</td>
</tr>
<tr>
<td>$F_8$</td>
<td>50</td>
<td>8.882E-16</td>
<td>8.882E-16</td>
<td>0.000E+00</td>
<td>100</td>
<td>8.882E-16</td>
<td>2.665E-15</td>
<td>1.794E-15</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>8.882E-16</td>
<td>8.882E-16</td>
<td>0.000E+00</td>
<td>100</td>
<td>8.882E-16</td>
<td>3.091E-15</td>
<td>1.742E-15</td>
</tr>
<tr>
<td>$F_9$</td>
<td>50</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>100</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>100</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>$F_{10}$</td>
<td>50</td>
<td>3.609E-17</td>
<td>3.879E-13</td>
<td>7.469E-13</td>
<td>100</td>
<td>1.564E-10</td>
<td>5.083E-03</td>
<td>1.703E-02</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>2.684E-16</td>
<td>1.362E-10</td>
<td>2.904E-10</td>
<td>100</td>
<td>1.606E-04</td>
<td>6.100E-03</td>
<td>8.900E-03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>D</th>
<th>Min</th>
<th>Mean</th>
<th>Std</th>
<th></th>
<th>Min</th>
<th>Mean</th>
<th>Std</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 6** Comparison of optimization results on uni-modal and multi-modal benchmark test functions($F_1 - F_{10}$)

<table>
<thead>
<tr>
<th>Function</th>
<th>D</th>
<th>Min</th>
<th>Mean</th>
<th>Std</th>
<th>D</th>
<th>Min</th>
<th>Mean</th>
<th>Std</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
These fixed-dimensional multi-modal test functions primarily assess the performance of algorithms in balancing exploration and exploitation capabilities. The optimization results are less affected by the variation in dimensions, and the HI-WHO outperforms the other five algorithms in terms of optimization accuracy. For the uni-modal test functions $F_1$ and $F_2$, as well as the multi-modal test functions $F_3$ and $F_4$, the minimum value, average value, and variance of the optimization results remain the theoretical optimal values across different dimensions, which indicates that the HI-WHO achieves excellent and stable performance.

Table 8 presents the comparative results of the six algorithms on fixed-dimensional multi-modal benchmark test functions. These fixed-dimensional multi-modal test functions primarily assess the performance of algorithms in balancing exploration and exploitation capabilities. From the data in the table, it can be observed that all six algorithms achieve optimization results close to the theoretical optimal values for functions $F_{11}$ to $F_{15}$. However, the HI-WHO consistently outperforms the other five algorithms in terms of the minimum value, average value, and standard deviation. Particularly for the function $F_{14}$, the HI-WHO achieves the theoretical optimal value for both the minimum and average values. This indicates that the introduction of dynamic nonlinear adaptive weight factors effectively balances the proposed algorithm’s global and local search performance.

The data in Table 8 represents the results obtained using the six algorithms to solve five complex global optimization problems from the CEC2022 test suites. These functions have been subject to be shifted and rotated, making it difficult to find their global optimum values. Solving these functions’ global optimum solutions in low dimensions makes it possible to evaluate the performance of metaheuristic algorithms effectively (W. Xu, Zhang, & Chen, 2022). From the data in the table, it can be observed that except for function $F_{19}$ where the HI-WHO ranks second in terms of the standard deviation (Std), surpassed only by the IWHOLF algorithm, the HI-WHO achieves the best results in the remaining test cases. The proposed algorithm demonstrates better performance in solving complex global optimization problems.

The above results and analysis demonstrate that, regardless of whether uni-modal or multi-modal, low-dimensional or high-dimensional, the HI-WHO proposed in this study exhibits superior optimization performance compared to the other five representative benchmark algorithms with better solving capabilities. The HI-WHO demonstrates excellent optimization accuracy...
To provide a more intuitive observation of the optimization performance of the algorithms, Figure 8 illustrates the convergence curves of the six algorithms when solving the complex standard test functions (F₁₁ – F₁₅) to demonstrate that HI-WHO converges to the optimal value with the fewest iterations and exhibits the strongest and stability, effectively addressing the problems of the standard WHO, such as susceptibility to local optima, instability in optimization performance, and occasional lack of high precision in solving complex function global optimization problems.

### 5.3.2 Analysis of convergence accuracy and algorithm stability

To provide a more intuitive observation of the optimization performance of the algorithms, Figure 8 illustrates the convergence curves of the six algorithms when solving the complex standard test functions F₁ to F₁₀ in D=100 dimensions and the fixed-dimensional benchmark test functions F₁₁ to F₁₅.

The convergence curve trends observed from the graphs indicate that HI-WHO outperforms the other five algorithms. It exhibits faster convergence, fewer instances of getting trapped in local optima, and higher optimization accuracy and capability. In the case of solving high-dimensional uni-modal and multi-modal problems, the convergence curves of functions F₃, F₆, F₇, F₈, and F₉ demonstrate that HI-WHO converges to the optimal value with the fewest iterations and exhibits the strongest...
optimization ability. Moreover, for functions $F_1$, $F_2$, $F_4$, $F_5$, and $F_{10}$, the HI-WHO shows higher optimization accuracy. When solving fixed-dimensional multi-modal problems, although HI-WHO and other algorithms can all be close to the theoretical optimum, the HI-WHO demonstrates the highest stability. The HI-WHO quickly converges to the theoretical optimum value in optimizing functions $F_{11}$ to $F_{15}$.

Figure 9 showcases box plots comparing the performance of the HI-WHO with five other algorithms in solving the CEC2022 benchmark functions ($F_{16} - F_{20}$). The plots reveal that all algorithms, including the HI-WHO, converge around the theoretical values. Notably, the HI-WHO exhibits the lowest standard deviation, indicating remarkable stability. The enhanced algorithm effectively mitigates the risk of falling into local optima and significantly improves the problem of premature convergence.

Additionally, Figure 10 illustrates the dynamic changes in the exploration and exploitation percentages of the HI-WHO during the optimization process of the CEC2022 benchmark functions. In population-based metaheuristic algorithms, assessing the behavior of individual solutions and the overall population is critical. By conducting dimensional diversity tests and quantifying the exploration and exploitation percentages at each iteration, the efficacy of the algorithm’s enhancement strategies can be thoroughly evaluated (Hussain, Salleh, Cheng, & Shi, 2019; Morales-Castañeda, Zaldivar, Cuevas, Fausto, & Rodríguez, 2020). As depicted in the figure, the HI-WHO successfully enhances its global exploration capability and local exploitation capacity by
optimizing the initial population distribution and incorporating dynamic nonlinear adaptive factors. This balances the trade-off between exploration and exploitation, effectively addressing premature convergence.

**Figure 9** Box plots of CEC2022 benchmark test functions $F_{16}$ to $F_{20}$.

**Figure 10** Exploration and exploitation of HI-WHO on CEC2022 benchmark test functions.
6  APPLICATION OF THE HI-WHO IN ROBOT PATH PLANNING

In order to verify the feasibility and effectiveness of the improved HI-WHO in solving robot path planning problems, this section conducts simulation experiments in simple and realistically abstracted grid map environments. The comparative experimental analysis includes the Improved PSO algorithm (Song et al., 2021), the ISSA (G. Zhang & Zhang, 2022b), the standard WHO, and the improved HI-WHO proposed in this paper.

The experiments were conducted on the same hardware platform and programming environment to ensure fairness and objectivity. The host hardware platform had an Intel(R) Core(TM) i7-8550U processor, 1.8GHz with the Windows 10 64-bit operating system, and 32GB of memory. The experiments were carried out using Matlab 2020a software. Under the same conditions, the population size for all four algorithms was set to $N = 30$, and the maximum number of iterations was set to $T_{\text{max}} = 200$. The other parameter settings for the comparative algorithms were consistent and are shown in Table 5.

6.1  Simulation experiment of robot path planning in simple environment

The simulation experiment of path planning in a simple environment is conducted on a grid map of $30 \times 30$. The robot starts from the circular symbol at coordinates (0.5, 0.5) and moves towards the target point represented by the pentagram symbol at coordinates (29.5, 29.5). The black areas in the map represent obstacles, while the white areas represent the robot’s free space for movement.

Four different algorithms are employed to perform robot path planning in this area. After 30 repeated experiments, the optimal path results are compared in Figure 11 and the corresponding fitness curve comparison is shown in Figure 12. The worst value, best value, average value, standard deviation, and turn times of the path lengths are calculated and summarized from the data collected in the 30 repeated experiments, as presented in Table 10.

From Figures 11 and 12, it can be observed that all four algorithms successfully avoid obstacles and complete the path planning from the start point to the target point. The Improved PSO algorithm converges to a fitness value of 43.24 after 49 iterations, the ISSA converges to a fitness value of 43.70 after 20 iterations, the standard WHO converges to a fitness value of 42.08 after 49 iterations, and the improved HI-WHO converges to a fitness value of 41.65 after 39 iterations. The fitness value represents the path length since none of the algorithms’ planned paths intersect with obstacles, according to Equation (3). Therefore, comparing the path planning results of the four algorithms, the HI-WHO achieves the optimal path length. Compared to the standard WHO, the HI-WHO generates shorter paths, reducing the number of algorithm iterations by 20%.

From the statistical results in Table 10, it can be observed that in the 30 repeated path planning experiments, the HI-WHO has the lowest average path length and the smallest standard deviation. The paths generated by the HI-WHO exhibit less variation between each planning iteration, indicating a higher stability level than the other three algorithms. Figure 13 illustrates the effect
A NOVEL ALGORITHM FOR ROBOT PATH PLANNING BASED ON HYBRID STRATEGY IMPROVED WILD HORSE OPTIMIZER

FIGURE 12 Convergence curves of Improved PSO, ISSA, WHO and HI-WHO.

FIGURE 13 Effect of path smoothing using B-Spline curve.

TABLE 10 Experimental results in simple environment

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Worst</th>
<th>Best</th>
<th>Average</th>
<th>Std</th>
<th>Turn times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improved PSO</td>
<td>46.87</td>
<td>43.24</td>
<td>45.62</td>
<td>0.46</td>
<td>10</td>
</tr>
<tr>
<td>ISSA</td>
<td>48.17</td>
<td>43.70</td>
<td>47.49</td>
<td>0.62</td>
<td>9</td>
</tr>
<tr>
<td>WHO</td>
<td>44.26</td>
<td>42.08</td>
<td>43.74</td>
<td>0.28</td>
<td>10</td>
</tr>
<tr>
<td>HI-WHO</td>
<td>42.14</td>
<td>41.65</td>
<td>41.36</td>
<td>0.12</td>
<td>8</td>
</tr>
</tbody>
</table>

of applying B-Spline curve smoothing to the path generated by the HI-WHO. The smoothed path exhibits smoother changes in the turning angles, effectively avoiding sharp turns or jittering, thereby improving the efficiency and stability of the robot’s movements.
6.2 Simulation experiment of robot path planning in realistically abstracted grid map environment

We conducted path-planning experiments on a real building floor structure which is using the grid-based representation of its environment, as shown in Figure 14, to further validate the effectiveness and feasibility of the improved HI-WHO in solving real-world path-planning problems.

![Figure 14: Floor plan of a real building structure.](image)

The experimental results are shown in Figures 15 and 16. Similarly, four algorithms were independently tested for 30 trials. The statistical analysis of the worst value, best value, average value, standard deviation, and number of turns for each algorithm are presented in Table 11.

![Figure 15: Optimal paths generated by Improved PSO, ISSA, WHO and HI-WHO.](image)

From 15 and 16, it can be observed that the improved HI-WHO achieves the best path planning results among the four algorithms. Specifically, after 100 iterations, the Improved PSO algorithm converges to the optimal value of 76.32, the ISSA converges to 75.40 after 87 iterations, the standard WHO converges to 74.71 after 58 iterations, and the improved HI-WHO
A NOVEL ALGORITHM FOR ROBOT PATH PLANNING BASED ON HYBRID STRATEGY IMPROVED WILD HORSE OPTIMIZER

Figure 16  Convergence curves of Improved PSO, ISSA, WHO and HI-WHO.

converges to 72.01 after 46 iterations. The improved HI-WHO benefits from the Sobol sequence initialization of the population, which enables it to quickly search within the vicinity of the optimal solution at the initial stage. Furthermore, the algorithm balances global exploration and local exploitation capabilities through adaptive inertia weight and lens imaging opposition-based learning strategy, enabling fine-grained search within the search space and achieving convergence to the optimal value faster. Compared to the standard WHO, the improved HI-WHO reduces the number of iterations by 14%. Additionally, the path generated by the improved HI-WHO exhibits smaller turning angles and better performance than the other three methods.

Table 11  Experimental results in simple environment

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Worst</th>
<th>Best</th>
<th>Average</th>
<th>Std</th>
<th>Turn times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improved PSO</td>
<td>80.72</td>
<td>76.32</td>
<td>78.76</td>
<td>0.69</td>
<td>10</td>
</tr>
<tr>
<td>ISSA</td>
<td>78.23</td>
<td>75.40</td>
<td>77.12</td>
<td>0.73</td>
<td>10</td>
</tr>
<tr>
<td>WHO</td>
<td>76.14</td>
<td>74.71</td>
<td>75.68</td>
<td>0.48</td>
<td>11</td>
</tr>
<tr>
<td>HI-WHO</td>
<td>72.54</td>
<td>72.01</td>
<td>72.27</td>
<td>0.16</td>
<td>8</td>
</tr>
</tbody>
</table>

The results from Table 11 demonstrate that the HI-WHO outperforms the other three algorithms regarding the worst value, best value, average value, standard deviation, and turn times of the path length. Additionally, the algorithm exhibits the best stability. The effectiveness of the HI-WHO in path planning can be further illustrated by the smoothed paths obtained through three iterations of B-Spline curve fitting, as shown in Figure 17 Overall, based on the comprehensive analysis of the experiments, the improved HI-WHO achieves the shortest path length and demonstrates superior performance compared to other algorithms. It satisfies the requirements of global path planning for mobile robots in complex environments.

7  CONCLUSION AND FUTURE WORKS

This work proposes a novel algorithm based on hybrid strategy improved Wild Horse Optimizer to enhance the capability of the standard Wild Horse Optimizer (WHO) and expand its application potential in solving global path planning problems for mobile robots. The proposed algorithm overcomes the shortcomings of the standard WHO by combining Sobol sequences for population initialization, as well as integrating Lévy flight strategy, dynamic self-adaptive factor, and lens imaging opposition-based learning strategy. These modifications increase the diversity of the population and balance the algorithm’s global exploration and local exploitation abilities.

Furthermore, a mathematical model for robot path planning problems is established, allowing the application of the improved HI-WHO to mobile robot path planning. Simulated experiments are carried out using grid-based maps in both simple and real-world environments. The results demonstrate that the improved HI-WHO achieves faster convergence, with iteration counts decreasing by 20% and 14% in the respective environments. Additionally, it exhibits superior comprehensive performance in
solving path planning problems, and the smoothed paths obtained through cubic B-Spline curves can meet the requirements of global path planning and motion control for robots.

However, the enhanced HI-WHO still exhibits some limitations despite the improvements. For instance, while solving specific functions in uni-modal and multi-modal problems, such as $F_3$, $F_6$, and $F_{10}$, it may still encounter problems with getting trapped in local optima. Additionally, the algorithm’s accuracy in solving CEC2022 benchmark suites is relatively low, indicating that there is room for further enhancement in algorithm performance. In the next steps, we will focus on refining the optimization capability and performance of the HI-WHO. The goal is to enhance its stability and adaptability, and apply it in solving optimization problems related to multi-robot path planning and other complex real-world scenarios. These improvements aim to address the existing limitations while contributing to the overall effectiveness and practicality of the algorithm.

**AUTHOR CONTRIBUTIONS**

Each author made significant contributions to this work. Juntao Zhao contributed to the initial version of the manuscript, the development and validation of the code, as well as the conceptualization and design of the study. Yong Li performed data analysis and interpretation. Xiaochuan Luo provided critical revisions and supervision throughout the research process. All authors have read and approved the final manuscript.

**ACKNOWLEDGMENTS**

This work was supported by National Key R&D Program of China (2019YFB1705002), National Natural Science Foundation of China (51634002), LiaoNing Revitalization Talents Program (XLYC2002041).

**CONFLICT OF INTEREST**

The authors declare that there are no conflict of interest in this paper.

**AVAILABILITY OF DATA AND MATERIALS**

The research conducted in this study did not involve the creation of new data. Therefore, no specific research data are available or shared.

**ORCID**

Xiaochuan Luo [https://orcid.org/0000-0001-6886-1649](https://orcid.org/0000-0001-6886-1649)

**References**


**AUTHOR BIOGRAPHY**

**Juntao Zhao** received the B.E. degree in electrical engineering and automation from Hebei University of Technology, China, in 2013 and the M.Eng. degree in control engineering from China University of Geosciences Beijing, China, in 2021. He is currently pursuing the doctoral degree with Northeastern University, Shenyang, China. He is also a member of laboratory EPROAD, Amiens, France, from 2022. His major research interests include the application of intelligent algorithms, trajectory online optimization, combinatorial optimization, mathematical modeling and reinforcement learning. He is also served as a peer reviewer for *International Journal of Robotics and Automation*.

**Xiaochuan Luo** received the M.Eng. and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1999 and 2002, respectively. From 2002 to 2004, he held a postdoctoral position at the University of Technology of Troyes, Troyes, France. He is currently a Professor with Northeastern University, Shenyang, China. His current research interests include modeling and optimization of processing industry manufacturing systems, production planning and scheduling, and optimization methods. He is a peer review expert of the National Natural Science Foundation of China, the IEEE Reviewer, IJPR, EJIE, and other international journal reviewers. He was selected into the 2008 Ministry of Educations New Century Excellent Talents Support Program and the Million Talents Project in Liaoning Province, in 2009.
Yong Li received the B.E. degree in new energy science and engineering from Xuzhou University of Technology, Xuzhou, China, in 2021 and he is currently pursuing the M.Sc. degree with the College of Information Science and Engineering, Northeastern University, Shenyang, China. His main research interests include control theory, path optimization, and intelligent algorithms.