IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 June 2023, accepted 24 June 2023, date of publication 27 June 2023, date of current version 10 July 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3289856*

RESEARCH ARTICLE

# Multi-Strategy Fusion Improved Adaptive Hunger Games Search

**DAMING ZHANG, YANQING ZHAO, JUNJIE DING, ZIJIAN WANG, AND JIAQING XU**
Guilin University of Technology, Guilin 541006, China

Corresponding author: Daming Zhang (zhangdm@glut.edu.cn)

This work was supported in part by the Project by the National Natural Science Foundation of China under Grant 52178468 and Grant 52268023, in part by the National Natural Science Foundation of Guangxi under Grant 2023GXNSFAA026418, in part by the Open Foundation of Guangxi Key Laboratory of Embedded Technology and Intelligence under Grant 2019-02-08, and in part by the Guangxi Key Laboratory of Geomechanics and Geotechnical Engineering under Grant GUIKENENG19-Y21-2.

**ABSTRACT** Aiming at the drawbacks of Hunger Games Search (HGS) algorithm, such as slow convergence speed and the tendency to fall into local optimum, a Multi-strategy fusion Improved Adaptive Hunger Games Search (MIA-HGS) algorithm is proposed. Firstly, a good point set is employed to generate a more diverse initial population. Secondly, the control strategy selection parameter is fixed in the original HGS algorithm; an adaptive adjustment parameter is proposed to replace the fixed parameters, whose dynamically tuned update strategy strengthens the global searching ability. Finally, to further jump out of the local optimum, a mutation operation based on Logarithmic spiral opposition-based learning is performed on a population for a certain condition. Simulation experiments are carried out for 23 benchmark functions and the UAV aerial planning problem. The results show that MIA-HGS solves more accurately and converges more rapidly than the original HGS algorithm on 23 benchmark functions, with MIA-HGS leading on 69.5% of the tested functions and tying with HGS on 21.7% of the tested functions. It also showed better performance than the other algorithms on the UAV flight planning problem.

**INDEX TERMS** Hunger games search, good point set, adaptive strategies, logarithmic spiral opposition-based learning, UAV aerial planning.

## I. INTRODUCTION

In solving optimization problems [1], swarm intelligent optimization algorithms have now become a research hotspot for scholars due to their simple structure and easy implementation [2], [3], [4]. In recent years, a large number of intelligent optimization algorithms based on population foraging have been introduced. For example, Particle Swarm Optimization algorithm (PSO) [5], Grey Wolf Optimizer algorithm (GWO) [6], Sparrow Search Algorithm (SSA) [7], Whale Optimization Algorithm (WOA) [8], etc.. These optimization algorithms have a similar structure and each has its own advantages, and disadvantages.

The Hunger Games Search algorithm (HGS) [9] is a swarm intelligence optimization algorithm proposed by Yang et al. in 2021 to simulate animal hunger foraging. The algorithm is not only a simulation of animal foraging behavior but also takes into account the physical and psychological factors of the animals. The algorithm is made to fit the behavioral logic of most animals foraging for food. It has been shown in the original literature [6] to be significantly better than the Particle Swarm Optimization algorithm (PSO), Differential Evolution algorithm (DE), Whale Optimization Algorithm (WOA) and Grey Wolf Optimizer algorithm (GWO) in terms of optimization accuracy and optimization stability. Due to its superior performance in finding the best solution, it has been applied in various fields such as wireless sensor networks [10], friction welding [11], solar photovoltaics [12], soil winding rate [13], cyber security [14], medical image processing [15] and Optimal Charging/Discharging Decision of Energy Storage [16].

Although HGS has good performance and has been used successfully in various fields, it also suffers from the common problems of group intelligence optimization algorithms, namely the tendency to fall into local optima and slow convergence. Since HGS has been proposed for a short period of time, only a few scholars have improved it. For example, literature [17] used different chaotic mappings to increase

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos.

population diversity at different times; literature [18] proposed an improved HGS algorithm with multi-strategy integration by exploiting the global search capability of Multi-Strategy (MS) framework; literature [19] combined HGS with WOA; literature [20] increased the ability of the algorithm to jump out of the local optimum through a logarithmic spiral inverse learning mechanism.

This study is dedicated to improving the shortcomings of the traditional HGS algorithm and enhancing the overall optimization capability of the algorithm. Therefore, this paper proposes a Multi-strategy fusion Improved Adaptive Hunger Games Search algorithm (MIA-HGS). In this paper, the original HGS algorithm is improved at three levels. Firstly, at the initial stage of the population, the good point set strategy [21] is used to enhance the initial population diversity; Secondly, the parameters are improved so that they have adaptive adjustment ability, thus different search strategies are selected to improve the search performance of the algorithm; Finally, stagnation detection and logarithmic spiral opposition-based learning mechanisms are introduced to increase the ability of the algorithm to jump out of local optimum.

## II. HUNGER GAMES SEARCH ALGORITHM (HGS)

The Hunger Games search algorithm is designed to simulate the activity behavior of herd animals during foraging. In nature, animals hide from natural predators, cooperate with each other or hunt alone, creating a natural law of survival of the fittest. In this brutal competitive relationship, animals inevitably have to improve themselves, i.e., evolve, in order to improve their chances of survival. The hungrier they are, the more they crave food and thus ensure their survival. At the same time, the stronger they are, the more likely they are to get food. This phenomenon is called nature's hunger game [22].

### A. APPROACH FOOD

The foraging process involves both teamwork and individual action, with the animals foraging in the following ways:

$$\overrightarrow{X(t+1)} = \begin{cases} Game_1 : \overrightarrow{X(t)} \cdot (1 + randn(1)), \quad r_1 < l \\ Game_2 : \overrightarrow{W_1} \cdot \overrightarrow{X_b} + \overrightarrow{R} \cdot \overrightarrow{W_2} \cdot \left| \overrightarrow{X_b} - \overrightarrow{X(t)} \right|, \\ \qquad r_1 > l, \quad r_2 > E \\ Game_3 : \overrightarrow{W_1} \cdot \overrightarrow{X_b} - \overrightarrow{R} \cdot \overrightarrow{W_2} \cdot \left| \overrightarrow{X_b} - \overrightarrow{X(t)} \right|, \\ \qquad r_1 > l, \quad r_2 < E \end{cases} \tag{1}$$

where $Game_1$ simulates animals foraging alone, i.e., looking for food near themselves; $Game_2$ and $Game_3$ simulate group cooperative foraging, looking for food around individuals in the population that have already found food (the best adapted); $\overrightarrow{R}$ is a random number between $[-a, a]$ and $a$ is a convergence factor that decreases linearly from 2 to 0 with increasing iterations; $r_1$ and $r_2$ are both random numbers between [0, 1] random numbers; $randn(1)$ is a normally distributed random number satisfying a mean of 0 and a standard deviation of 1; $\overrightarrow{W_1}$ and $\overrightarrow{W_2}$ denote hunger weights, which are adjusted to control foraging based on the influence of environmental and psychological factors; $\overrightarrow{X_b}$ denotes the optimal global position; $t$ represents the number of current iterations; $\overrightarrow{X(t)}$ denotes the current individual position; the parameters $l = 0.08$; $E$ is calculated as follows:

$$E = \operatorname{sech}\left(|F(i) - BF|\right) \tag{2}$$

where $F(i)$ denotes the fitness of each individual; $BF$ is the best fitness obtained during the current iteration; the sech() is a hyperbolic function.

$$\operatorname{sech}(x) = \frac{2}{e^x + e^{-x}} \tag{3}$$

$\overrightarrow{R}$ is calculated as follows:

$$\overrightarrow{R} = 2 \times a \times r_3 - a \tag{4}$$

$$a = 2 \times \left(1 - \frac{t}{T}\right) \tag{5}$$

where $r_3$ is a random number between [0, 1]; $T$ represents the maximum number of iterations.

### B. HUNGER ROLE

The hunger characteristics of individuals in the search were mathematically modelled. The formula for $\overrightarrow{W_1}$ in Eq. (1) is as follows:

$$\overrightarrow{W_1(i)} = \begin{cases} hungry(i) \cdot \dfrac{N}{SHungry} \times r_4, & r_3 < l \\ 1, & r_3 > l \end{cases} \tag{6}$$

The formula for $\overrightarrow{W_2}$ in Eq. (1) is as follows:

$$\overrightarrow{W_2(i)} = \left[1 - \exp\left(-\left|hungry(i) - SHungry\right|\right)\right] \times r_5 \times 2 \tag{7}$$

where $hungry$ represents the hunger level of each individual; $N$ is the total number of individuals; $SHungry$ is the sum of hunger levels of all individuals, i.e. $sum\,(hungry)$; $r_3$, $r_4$ and $r_5$ are all random numbers between [0, 1]; $hungry(i)$ is calculated as follows:

$$hungry(i) = \begin{cases} 0, & F(i) = BF \\ hungry(i) + H, & F(i) \neq BF \end{cases} \tag{8}$$

In each iteration, the hunger of the individual closest to the food (with the best adaptation) is 0. $H$ is calculated as follows:

$$H = \begin{cases} LH \times (1 + r_6), & TH < LH \\ TH, & TH \geq LH \end{cases} \tag{9}$$

$$TH = \frac{F(i) - BF}{WF - BF} \times r_7 \times 2 \times (UB - LB) \tag{10}$$

where $r_6$ and $r_7$ are both random numbers between [0, 1]; $LH$ is the lower bound of $H$, taken as $LH = 100$; $WF$ is the worst fit obtained during the previous iteration; $UB$ and $LB$ denote the upper and lower bounds of the search space respectively.

## III. ALGORITHM IMPROVEMENT STRATEGIES

### A. GOOD POINT SET STRATEGY

In the original HGS algorithm, the initialized population is generated randomly within the search range, which has a large uncertainty and makes it difficult to guarantee the diversity of the population. This will directly lead to a significant reduction in the convergence speed of the algorithm. In contrast, the good point set [23] solves this problem to the fullest extent possible, and the initial population it generates is more evenly distributed within the search range, which greatly improves the population diversity. It has been shown in the literature [24], [25], [26] that good point sets can effectively improve the convergence speed of the algorithm. Therefore, this paper adopts the good point set to enhance the diversity of the initial population.

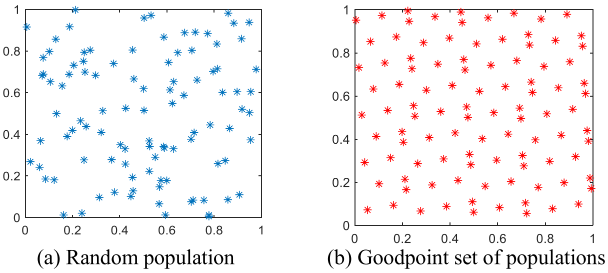
Good point set expressions:

$$P_m\left(i\right) = \left\{\left(p_1, p_2, \cdots, p_m\right) \times i\right\}, \quad i = 1, 2, \cdots, n \tag{11}$$

where $p$ stands for good point, $p_k = \bmod\left(2\cos\left(\frac{2\pi k}{s}\right), 1\right)$, $k = 1, 2, \cdots, m$. $n$ indicates the number of populations, mod () is a remainder function, which means that $p_i$ takes only the fractional part of $2\cos\left(\frac{2\pi k}{s}\right)$, $s$ is the smallest prime number that satisfies $\frac{s-3}{2} \geq m$; $m$ represents the spatial dimension of the population search. When initializing the population, it is sufficient to map the set of good points to the search space:

$$x_{ij} = LB_j + P_m\left(i\right)\left(UB_j - LB_j\right) \tag{12}$$

where $UB_j, LB_j$ denote the upper and lower search bounds of the $j$ dimension. In order to more intuitively demonstrate the effect of using the good point set strategy to initialize the population, this paper uses the random strategy and the good point set strategy to generate 100 discrete points in a 1 × 1 2D space respectively, and the comparison effect is shown in Fig. 1. As can be seen from Fig. 1, the randomly generated discrete points are messy and some of them even overlap. However, with the good point set strategy, each discrete point was evenly distributed in space and no overlap was found, which greatly improved the diversity of the population and laid a good foundation for the next iterative search.



(a) Random population (b) Goodpoint set of populations

**FIGURE 1. Comparison chart of initialized populations.**

### B. ADAPTIVE STRATEGY ADJUSTMENT

The parameters in the central equation of the HGS (which is Eq. (1)) control the choice of strategy. In the original HGS, it is common to take $l = 0.08$, such that the probability of an animal foraging alone is only 8%, which means that 92% of the individuals in the population search around the optimal individual from the beginning, completely discarding their previous position. This leads to premature convergence of the algorithm, which directly loses the diversity of the population and increases the probability of the algorithm falling into a local optimum. Although the original algorithm with a parameter of 0.08 was also experimentally discussed, the maximum value of the parameter in the experiment was only 0.1.

In this study, we will improve the parameter $l$ and propose an adaptive adjustment strategy: in the early stage of the algorithm, we will continue to maintain the population diversity, so that the population will search more in its own vicinity, providing a good basis for the later search and reducing the probability of falling into the local optimum; in the middle stage of the algorithm, we should strengthen the cooperation between the populations, i.e. search near the optimal individuals of the population to form a fast convergence in the local area In the middle of the algorithm, we need to strengthen the inter-population cooperation, i.e. to search around the optimal individuals of the population to form a fast convergence in the local area, so that the accuracy of the algorithm can be improved rapidly; in the later stage, the algorithm has reached a certain accuracy, in order to further improve the accuracy of the algorithm and to find whether there are other extreme value points, the individual search should be mainly used to find a better extreme value point.

In summary, five different parameter adjustment strategies based on the normal distribution, power function, and Cauchy distribution are constructed in this paper, and the comparison graph is shown in Fig. 2. After experimental testing and verification, the parameter adjustment strategy based on the normal distribution was finally chosen, with the parameter $l$ changing with the number of iterations $t$, i.e.:

$$l = b - \frac{1}{\sqrt{2\pi}\sigma}\exp\left(\frac{\left(\frac{t}{T} - \mu\right)^2}{2\sigma^2}\right) \tag{13}$$

where $T$ is the maximum number of iterations set by the algorithm, the algorithm has been tested to find the best results when $\mu = 0.5$, $\sigma^2 = 0.45$, $b = 0.8$ are taken.

### C. LOGARITHMIC SPIRAL OPPOSITION-BASED LEARNING

In order to further enhance the ability of the algorithm to jump out of the local optimum and, at the same time, speed up the convergence rate of the algorithm. When the population optimal fitness value does not change in a certain number of population iterations, the algorithm is judged to have entered a stagnant state, at which point the logarithmic spiral opposition-based learning mechanism is introduced. Although it may lead to an increase in the algorithm's time complexity, the conditions under which this mechanism occurs are mostly in the middle to late stages of the algorithm, i.e. the algorithm has largely confirmed the optimal solution.

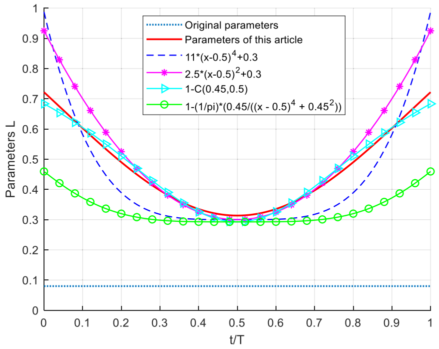


**FIGURE 2. Comparison chart of different strategies for parameter *l*.**

However, it may be a locally optimal solution, so appropriate variation in the population is required to make the algorithm jump out of the local optimum.

The logarithmic spiral opposition-based learning mechanism is based on opposition-based learning (OBL) [27], incorporating logarithmic spirals.

OBL maps the current solution $X$ to the opposite of the feasible domain $[UB, LB]$, i.e.. $X_{op} = UB + LB - X$. This causes the current solution $X$ to change radically, greatly improving the ability of the algorithm to jump out of the local optimum. This relatively extreme approach, however, only mutates the current solution $X$ in a fixed direction, and although it jumps out of the local optimum, it does not necessarily facilitate the search of the algorithm at a later stage. Therefore some randomness needs to be added so that the algorithm takes into account more regions, i.e.:

$$X_{op} = r_8 \times UB + r_9 \times LB - X_b \tag{14}$$

where $r_8$, $r_9$ is a random number between 0 and 1 and is the global optimal solution. Secondly, the search area is further extended by fusing the logarithmic helix on the basis of Eq. (14).

$$X_{new} = \left|X_b - X_{op}\right| \times e^k \times \cos(2\pi k) \tag{15}$$

In Eq. (15) $k = 2 \times r_{10} - 1$, $r_{10}$ is a random number between 0 and 1.

Finally, although logarithmic spiral OBL can produce changes in the population, such updates do not necessarily move in a good direction, so this paper adopts the idea of greed, retaining only those solutions that have improved fitness after a position change and not recording those that move in a worse direction.

$$X(t) = \begin{cases} X_{new}(t), & f(X_{new}(t)) < f(X(t)) \\ X(t), & f(X_{new}(t)) > f(X(t)) \end{cases} \tag{16}$$

### D. ALGORITHM STEPS

The steps of the Multi-strategy fusion Improved Adaptive Hunger Games Search algorithm(MIA-HGS) are as follows:

*Step 1:* Setting the basic parameters of the MIA-HGS;

*Step 2:* Initialize the population $X_1, X_2, \cdots, X_N$ using the good point set strategy, Eq. (11)(12);

*Step 3:* Calculate the fitness of each individual of the population according to the fitness search function, update and record the population optimal fitness $BF$ and $X_b$;

*Step 4:* Calculate the strategy selection parameter $l$ according to Eq. (13), calculate the strategy selection parameter $E$ according to Eq. (2)(3), $R$ calculate the convergence factor according to Eq. (4)(5) Calculate the starvation weights $W_1$ and $W_2$ according to Eq. (6)-(10), and finally update the population location according to Eq. (1);

*Step 5:* When the population optimal fitness $BF$ in Step 3 has not been updated for five or more times, a logarithmic spiral OBL mechanism will be introduced to update the variation for all individuals in the population, i.e. equations (14)-(16);

*Step 6:* If the current number of iterations $t$ exceeds the set maximum number of iterations then proceed to Step 7, otherwise jump back to Step 3;

*Step 7:* End of MIA-HGS algorithm, output optimal adaptation $BF$ and optimal position $X_{best}$.

The MIA-HGS algorithm flow is shown in Fig. 3.

## IV. SIMULATION EXPERIMENTS AND RESULTS ANALYSIS

In order to test the optimization-seeking performance of the MIA-HGS algorithm, 23 benchmark test functions from the literature [9] are selected for experimental comparison in this paper, and the Wilcoxon sign-rank test [28] were applied to identify the algorithms' significant differences. These 23 benchmark test functions are used in almost all articles on swarm intelligence optimization algorithms. The 23 benchmark test functions are shown in Tab. 1, where $F_1 - F_7$ is a high-dime national single-peaked function, $F_8 - F_{13}$ is a high-dimensional multi-peaked function, $F_{14} - F_{23}$ is a fixed-dimensional multi-peaked function. The Wilcoxon sign-rank test was used as a statistical significance test and when the result was '+' it indicated that MIA-HGS performed significantly better than its competitors.

The experimental environment was an AMD Ryzen 7 5800X CPU @ 3.80 GHz, 16.00 GB of RAM, Windows 11, and Matlab R2018a. All the algorithms selected in this paper used the same parameter settings, i.e., the number of populations was 30, the maximum number of iterations was 500, and the algorithms were run 30 times independently. The average and standard deviation of these 30 independent experiments were taken as the evaluation index of the algorithm's optimization-seeking simulation experiments.

### A. COMPARISON WITH CLASSICAL OPTIMIZATION ALGORITHMS

The MIA-HGS proposed in this paper is compared with some classical swarm intelligence optimization algorithms. The Hunger Games Search Algorithm (HGS), Particle Swarm Optimization Algorithm (PSO), Grey Wolf Optimizer Algorithm (GWO), Whale Optimization Algorithm (WOA), and Sparrow Search Algorithm (SSA) are selected in this paper. The parameters of the HGS algorithm are set as follows: $l = 0.08$, $LH = 100$. The parameters of the SSA algorithm

**TABLE 1.** 23 benchmark functions.

| Type | ID | Function Equation | Dim | Range | $f_{\min}$ |
|---|---|---|---|---|---|
| High dimensional Single-peak | F1 | $F_1(x)=\sum_{i=1}^{n} x_i^2$ | 30 | [-100,100] | 0 |
| | F2 | $F_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$ | 30 | [-10,10] | 0 |
| | F3 | $F_3(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i}x_j)^2$ | 30 | [-100,100] | 0 |
| | F4 | $F_4(x)=\max\{\lvert x_i\rvert,1\le i\le n\}$ | 30 | [-100,100] | 0 |
| | F5 | $F_5(x)=\sum_{i=1}^{n-1}[100(x_{i-1}-x_i^2)^2+(x_i-1)^2]$ | 30 | [-30,30] | 0 |
| | F6 | $F_6(x)=\sum_{i=1}^{n}(x_i+0.5)^2$ | 30 | [-100,100] | 0 |
| | F7 | $F_7(x)=\sum_{i=1}^{n} ix_i^4+random[0,1]$ | 30 | [-1.28,1.28] | 0 |
| High dimensional Multi-peak | F8 | $F_8(x)=\sum_{i=1}^{n}-x_i\sin(\sqrt{\lvert x_i\rvert})$ | 30 | [-500,500] | -12569.5 |
| | F9 | $F_9(x)=\sum_{i=1}^{n}\lvert x_i^2-10\cos(2\pi x_i)+10\rvert$ | 30 | [-5.12,5.12] | 0 |
| | F10 | $F_{10}(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$ | 30 | [-32,32] | 0 |
| | F11 | $F_{11}(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$ | 30 | [-600,600] | 0 |
| | F12 | $F_{12}(x)=\frac{\pi}{n}\{10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})+(y_n-1)^2]\}+\sum_{i=1}^{n}u(x_i,10,100,4)$ | 30 | [-50,50] | 0 |
| | F13 | $F_{13}(x)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2[1+\sin^2(3\pi x_i)]+(x_n-1)^2[1+\sin^2(2\pi x_n)]\}+\sum_{i=1}^{n}u(x_i,5,100,4)$ | 30 | [-50,50] | 0 |
| Fixed dimensions Multi-peak | F14 | $F_{14}(x)=(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{\sum_{i=1}^{n}(x_i-a_j)^6})^{-1}$ | 2 | [-65,65] | 1 |
| | F15 | $F_{15}(x)=\sum_{i=1}^{11}[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}]^2$ | 4 | [-5,5] | 0.0003 |
| | F16 | $F_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$ | 2 | [-5,5] | -1.0316 |
| | F17 | $F_{17}(x)=(x_2-\frac{5.1}{4\pi^2}+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos x_1+10$ | 2 | [-5,5] | 0.398 |
| | F18 | $F_{18}(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]$<br>$[30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$ | 2 | [-2,2] | 3 |
| | F19 | $F_{19}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{n}a_{ij}(x_j-p_{ij})^2)$ | 3 | [0,1] | -3.8628 |
| | F20 | $F_{20}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{n}a_{ij}(x_j-p_{ij})^2)$ | 6 | [0,1] | -3.32 |
| | F21 | $F_{21}(x)=-\sum_{i=1}^{5}\left[(x-a_i)(x-a_i)^{\mathrm{T}}+c_i\right]^{-1}$ | 4 | [0,10] | -10.1532 |
| | F22 | $F_{22}(x)=-\sum_{i=1}^{7}\left[(x-a_i)(x-a_i)^{\mathrm{T}}+c_i\right]^{-1}$ | 4 | [0,10] | -10.4028 |
| | F23 | $F_{23}(x)=-\sum_{i=1}^{10}\left[(x-a_i)(x-a_i)^{\mathrm{T}}+c_i\right]^{-1}$ | 4 | [0,10] | -10.5363 |

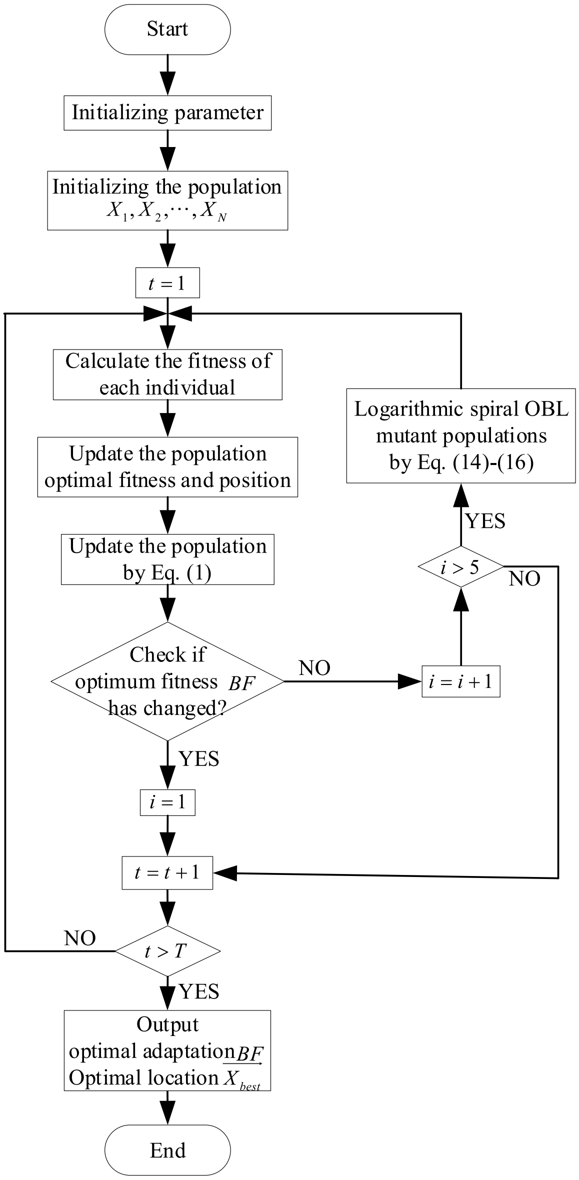

Start
Initializing parameter
Initializing the population $X_1, X_2, \cdots, X_N$
$t=1$
Calculate the fitness of each individual
Update the population optimal fitness and position
Update the population by Eq. (1)
Check if optimum fitness $BF$ has changed?
NO
$i=i+1$
$i>5$
YES
NO
Logarithmic spiral OBL mutant populations by Eq. (14)-(16)
YES
$i=1$
$t=t+1$
NO
$t>T$
YES
Output optimal adaptation $BF$ Optimal location $X_{best}$
End

**FIGURE 3.** MIA-HGS algorithm flow chart.

are set as in the literature [7]: $ST = 0.8$, $SD = 0.1$, $PD = 0.2$. The rest of the algorithm data were obtained from the literature [29]. A comparison of the algorithm data is shown in Tab. 2.

As can be seen from Tab. 2, the MIA-HGS has a higher search accuracy than the HGS in 69.5% of the tested functions and is equal to the HGS in 21.7% of the tested functions. Firstly, the high-dimensional single-peaked function $F_1 - F_7$ has only one extreme value point, which mainly tests the local fast convergence ability of the algorithm. MIA-HGS is equal to HGS only in $F_1$, and the rest are ahead. The above results prove that MIA-HGS has improved the fast local convergence performance compared with HGS; Secondly, there are multiple local optima on the high-dimensional multi-peak functions $F_8 - F_{13}$, which mainly test the algorithm's global search and ability to jump out of the local optimum. MIA-HGS and HGS can find the global optimum every time on $F_9 - F_{11}$, and there is no more room for improvement, so the results are equal. However, MIA-HGS improves on all the remaining test functions. The above results test demonstrates that MIA-HGS has enhanced its ability to search globally and jump out of local optima compared to HGS; Finally, in the fixed dimensional multi-peak function $F_{14} - F_{23}$, due to its low dimensionality and simple operation, MIA-HGS does not play its full performance and leads HGS in $F_{14} - F_{15}$ and $F_{19} - F_{23}$, whereas $F_{17} - F_{18}$ has the same finding accuracy as HGS, but still has a very small difference in standard deviation. Meanwhile the Wilcoxon sign-rank test showed that MIA-HGS was significantly better than HGS on 60% of the tested functions and only 17% of the tested functions were inferior to HGS. In summary, it fully demonstrates that the improved algorithm proposed in this paper is real and effective.

Compared to these five classical swarm intelligence optimization algorithms, MIA-HGS ranked first in terms of optimization results on 82.6% of the tested functions; tied with individual algorithms in terms of optimization accuracy on 8.6% of the tested functions, but with a slightly lower standard deviation; and ranked second on 4.3% of the tested functions. By the Wilcoxon sign-rank test, MIA-HGS was ahead of PSO, GWO and WOA across the board, although the disadvantage was most pronounced when comparing with SSA, which had only five cases. It can be concluded that MIA-HGS has some advantages over the classical swarm intelligence optimization algorithm.

### B. COMPARISON WITH IMPROVED HGS ALGORITHM

As HGS was proposed in 2021, there are fewer relevant improvements, and researchers have chosen different test functions, so this paper collects and replicates experiments as much as possible. This will be done with the Chaotic Hunger Games Search algorithm(Chaotic-HGS) [17], the Multi-Strategy ensemble Hunger Games Search algorithm(MS-HGS) [17], the Logarithmic spiral Opposition-Based Learning Hunger Games Search algorithm (LsOBL-HGS) [20], and the Elite Opposition-Based Learning and t-Distribution Hunger Games Search algorithm (EtHGS) [22]. The experimental data of Chaotic-HGS is directly derived from Scenario 2, which has the best improvement in the literature [17]. The experimental data for MS-HGS were obtained directly from the literature [18], the experimental data for LsOBL-HGS were derived by replicating the literature [20] with parameter $l = 0.08$, and the experimental data for EtHGS were mostly derived from the literature [22] with the remainder is derived from the experimental replication with parameter $l = 0.03$, $p = 0.8$.

Tab. 3 shows the experimental results of MIA-HGS and the four improved HGS algorithms. Among them, MIA-HGS is ranked first on $F_1 - F_4$, $F_6 - F_{11}$, $F_{13}$ and $F_{19}$; the average of the search for excellence on $F_{17} - F_{18}$ and $F_{21} - F_{23}$ has reached the optimum, only slightly inferior to the first

**TABLE 2.** Comparison of MIA-HGS and classical optimization algorithms for finding the best.

| Type | ID | MIA-HGS | | HGS | | PSO | | GWO | | WOA | | SSA | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | AGV | STD | AGV | STD | AGV | STD | AGV | STD | AGV | STD | AGV | STD |
| High dimensional Single-peak | F1 | 0 | 0 | 0 | 0 | 0.4615 | 0.2313 | 1.83e-27 | 2.98e-27 | 1.24e-72 | 5.60e-72 | 1.05e-39 | 1.05e-39 |
| | F2 | 0 | 0 | 8.06e-155 | 4.41e-154 | 4.9349 | 2.3901 | 9.70e-17 | 1.25e-16 | 4.72e-52 | 1.87e-51 | 8.21e-24 | 4.28e-23 |
| | F3 | 0 | 0 | 6.31e-197 | 0 | 50.0024 | 19.0070 | 1.81e-05 | 3.91e-05 | 4.61e+04 | 1.52e+04 | 1.40e-32 | 7.69e-32 |
| | F4 | 0 | 0 | 7.36e-140 | 4.03e-139 | 4.4878 | 1.7191 | 9.89e-07 | 1.00e-06 | 41.7900 | 29.2463 | 2.67e-25 | 1.46e-24 |
| | F5 | 1.7176 | 6.5326 | 10.9491 | 12.7327 | 257.6737 | 161.4050 | 26.7547 | 0.8234 | 27.7930 | 0.4673 | 3.37e-05 | 7.09e-05 |
| | F6 | 5.36e-07 | 1.97e-05 | 1.93e-05 | 3.15e-05 | 0.5019 | 0.2348 | 0.8148 | 0.4207 | 0.3641 | 0.2187 | 6.05e-07 | 2.85e-07 |
| | F7 | 1.05e-04 | 1.06e-04 | 4.72e-04 | 1.01e-03 | 0.0565 | 0.0563 | 0.0021 | 0.0014 | 0.0045 | 0.0050 | 2.84e-04 | 2.24e-04 |
| High dimensional Multi-peak | F8 | -12569.27 | 0.31655 | -12567.45 | 3.3426 | -2.71e+03 | 259.2516 | -5.90e+03 | 886.0377 | -1.04e+04 | 1.71e+03 | -8326.41 | 2906.43 |
| | F9 | 0 | 0 | 0 | 0 | 80.3380 | 20.1412 | 2.7689 | 4.4496 | 0 | 0 | 0 | 0 |
| | F10 | 8.88e-16 | 0 | 8.88e-16 | 0 | 5.2009 | 1.2190 | 1.02e-13 | 2.18e-14 | 5.15e-15 | 1.95e-15 | 3.49e-15 | 1.25e-14 |
| | F11 | 0 | 0 | 0 | 0 | 0.5854 | 0.1884 | 0.0036 | 0.0052 | 0 | 0 | 0 | 0 |
| | F12 | 4.56e-07 | 8.86e-07 | 1.06e-06 | 1.65e-06 | 3.3185 | 1.3489 | 0.0424 | 0.0181 | 0.0202 | 0.0141 | 1.75e-06 | 3.18e-08 |
| | F13 | 2.89e-06 | 4.87e-06 | 1.18e-05 | 2.25e-05 | 15.5409 | 15.6768 | 0.6691 | 0.2590 | 0.4736 | 0.2010 | 3.03e-06 | 8.04e-07 |
| Fixed dimensions Multi-peak | F14 | 1.3233 | 1.5829 | 3.2765 | 4.2008 | 2.5778 | 2.1297 | 5.2037 | 4.5541 | 2.9328 | 2.8852 | 11.0217 | 3.8242 |
| | F15 | 3.17e-04 | 5.06e-05 | 7.97e-04 | 2.54e-04 | 0.0027 | 0.0061 | 0.0057 | 0.0090 | 7.74e-04 | 5.62e-04 | 3.29e-04 | 3.44e-05 |
| | F16 | -1.0316 | 1.58e-08 | -1.0316 | 6.77e-16 | -1.0316 | 1.44e-16 | -1.0316 | 1.98e-08 | -1.0316 | 5.72e-10 | -1.0316 | 5.21e-16 |
| | F17 | 0.39789 | 2.14e-14 | 0.39789 | 0 | 0.39789 | 0 | 0.39789 | 1.28e-06 | 0.39789 | 1.75e-05 | 0.39789 | 6.54e-10 |
| | F18 | 3 | 6.55e-13 | 3 | 3.82e-15 | 3 | 3.67e-15 | 3 | 3.48e-05 | 3.0001 | 1.43e-04 | 3 | 6.61e-15 |
| | F19 | -3.8628 | 2.31e-19 | -3.8628 | 2.71e-15 | -3.8612 | 0.0032 | -3.8612 | 0.0027 | -3.8303 | 0.1402 | -3.7855 | 0.23587 |
| | F20 | -3.2544 | 6.00e-02 | -3.2505 | 5.93e-02 | -3.2223 | 0.1162 | -3.2449 | 0.1067 | -3.2324 | 0.1054 | -3.226 | 0.0969 |
| | F21 | -10.1532 | 1.50e-05 | -9.8133 | 1.2934 | -8.5143 | 2.6176 | -9.6460 | 1.5416 | -8.2059 | 3.0701 | -8.3779 | 2.344 |
| | F22 | -10.4029 | 3.99e-05 | -10.3924 | 5.73e-02 | -6.7470 | 3.7889 | -10.4011 | 9.44e-04 | -8.2711 | 3.1344 | -8.3037 | 2.6186 |
| | F23 | -10.5364 | 4.30e-05 | -10.1759 | 1.372 | -8.0138 | 3.5625 | -10.5347 | 8.70e-04 | -8.5819 | 3.0097 | -9.8001 | 1.8649 |
| Wilcoxon | +/−/= | ~ | | 14/4/5 | | 21/0/2 | | 21/0/2 | | 23/0/0 | | 10/5/8 | |

ranked algorithm in terms of standard deviation; and ranked second on $F_{12}$ and $F_{14}-F_{15}$. The average ranking results (the smaller the average ranking, the better the performance) are MIA-HGS < EtHGS < Chaotic-HGS < LsOBL-HGS < MS-HGS. The Wilcoxon sign-rank test experimental results also show that MIA-HGS has the advantage over other algorithms to find and develop the target solution space more quickly. In summary, MIA-HGS has the most comprehensive performance among the algorithms mentioned in this paper.

In order to more intuitively demonstrate the effect of the MIA-HGS and the other four HGS improvement algorithms, as well as the original HGS algorithm, in terms of finding the best results, this paper selects two of each of the three types from the 23 basic test functions, for a total of six test functions for iteration curve demonstration. Since the six algorithms converge in 50 generations, the paper will set the population size to 30, and the number of iterations to 50, and the effect is shown in Figure 4. From the iteration curves in Fig. 4, we can see that the starting point adaptation of MIA-HGS is relatively low, which fully reflects that the good point set strategy generates initial points with great population diversity and provides a good basis for the convergence of the later iterations of the algorithm. The starting point of MIA-HGS is not the lowest in $F_1$, $F_3$ and $F_{18}$ of Fig. 4, but it can catch up in the subsequent iterations. In the convergence comparison plot in Fig. 4, the MIA-HGS convergence curve is largely at the below, fully reflecting its speed advantage of iterative search.

## V. MIA-HGS APPLIED TO UAV 3D PATH PLANNING

In order to verify the effectiveness of MIA-HGS when solving practical problems, MIA-HGS is applied to UAV 3D path planning. The UAV 3D path planning problem is a superiority-seeking constraint problem. In the course of a mission, the UAV needs to reach a specified location from an initial position. When planning the path, factors such as terrain and obstacles need to be considered to avoid collisions between the UAV and the mountain peaks while keeping the path length as short as possible.

**TABLE 3.** Comparison of MIA-HGS with the latest improved optimization algorithm for finding the best.

| Type | ID | MIA-HGS | | Chaotic-HGS | | MS-HGS | | LsOBL-HGS | | EtHGS | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | AGV | STD | AGV | STD | AGV | STD | AGV | STD | AGV | STD |
| High dimensional Single-peak | F1 | 0 | 0 | 0 | 0 | 0 | 0 | 6.53e-38 | 3.21e-37 | 0 | 0 |
| | F2 | 0 | 0 | 0 | 0 | 2.81e-186 | 0 | 9.72e-23 | 5.22e-22 | 9.89E-237 | 0 |
| | F3 | 0 | 0 | 0 | 0 | 0 | 0 | 2.07e-05 | 1.13e-04 | 0 | 0 |
| | F4 | 0 | 0 | 0 | 0 | 2.00e-172 | 0 | 2.28e-16 | 1.11e-15 | 2.22E-221 | 0 |
| | F5 | 1.7176 | 6.5326 | 9.27 | 11.5 | 25.2 | 9.34 | 5.49e-03 | 0.02064 | 25.0269 | 0.61717 |
| | F6 | 5.36e-07 | 1.97e-05 | 3.17e-05 | 5.52e-05 | 1.68e-05 | 1.84e-05 | 7.77e-05 | 9.67e-05 | 4.64E-06 | 5.63E-06 |
| | F7 | 1.05e-04 | 1.06e-04 | 1.17e-04 | 1.35e-04 | 2.60e-03 | 3.20e-03 | 2.04e-03 | 2.07e-03 | 1.69e-04 | 2.16e-04 |
| High dimensional Multi-peak | F8 | -12569.27 | 0.31655 | -12568.20 | 1.9506 | -1.16e+04 | 4.89e+01 | -1.25e+04 | 250.0518 | -12294.39 | 522.5634 |
| | F9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | F10 | 8.88e-16 | 0 | 8.88e-16 | 0 | 8.88e-16 | 0 | 8.88e-16 | 0 | 8.88E-16 | 0 |
| | F11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | F12 | 4.56e-07 | 8.86e-07 | 4.13e-06 | 7.55e-06 | 3.86e-07 | 3.72e-07 | 2.50e-06 | [illegible]1e-06 | 1.33e-02 | 1.05e-02 |
| | F13 | 2.89e-06 | 4.87e-06 | 4.81e-02 | 0.2633 | 2.72e-05 | 6.22e-05 | 2.45e-05 | 4.01e-05 | 1.70e-02 | 9.33e-02 |
| Fixed dimensions Multi-peak | F14 | 1.3233 | 1.5829 | 1.3235 | 1.7829 | 1.97e+00 | 2.98e+00 | 0.998 | 2.70e-16 | 2.1729 | 2.9739 |
| | F15 | 3.17e-04 | 5.06e-05 | 3.28e-04 | 8.65e-05 | 5.40e-04 | 3.20e-04 | 5.64e-04 | 2.35e-04 | 4.4e-04 | 2.15e-04 |
| | F16 | -1.0316 | 1.58e-08 | -1.0316 | 2.83e-11 | -1.0316 | 8.75e-04 | -1.0316 | 5.68e-16 | -1.031 | 6.39e-16 |
| | F17 | 0.39789 | 2.14e-14 | 0.39789 | 1.78e-10 | 3.97e-01 | 0 | 0.39789 | 0 | 0.39789 | 0 |
| | F18 | 3 | 6.55e-13 | 3 | 2.31e-08 | 3.05e+00 | 1.11e-01 | 3 | 5.34e-15 | 3 | 0 |
| | F19 | -3.8628 | 2.31e-19 | -3.8627 | 1.98e-04 | -3.86e+00 | 4.62e-19 | -3.837 | 0.14113 | -3.862 | 2.65e-15 |
| | F20 | -3.2544 | 6.00e-02 | -3.217 | 8.89e-02 | -3.32e+00 | 1.45e-03 | -3.2863 | 0.05541 | -3.298 | 4.83e-02 |
| | F21 | -10.1532 | 1.50e-05 | -9.98 | 9.31e-01 | -9.84e+00 | 1.69e+00 | -10.1532 | 2.41e-14 | -10.1532 | 5.78e-15 |
| | F22 | -10.4029 | 3.99e-05 | -10.4 | 7.38e-16 | -1.02e+01 | 9.70e-01 | -10.4029 | 1.42e-13 | -10.4029 | 1.75e-15 |
| | F23 | -10.5364 | 4.30e-05 | -10.05 | 1.28e-15 | -1.04e+01 | 1.87e+00 | -10.5364 | 1.92e-13 | -10.5364 | 2.16e-15 |
| Wilcoxon | +/−/= | ~ | | 15/0/8 | | 17/5/1 | | 11/7/5 | | 12/8/3 | |

### A. TOPOGRAPHIC AREA MODEL

Ada The mathematical model for mountain terrain in 3D path planning is as follows [30]:

$$z_h(x,y) - \sum_{i=1}^{n} h_i \exp\left[-\left(\frac{x-x_i}{x_{si}}\right)^2 - \left(\frac{y-y_i}{y_{si}}\right)^2\right] \tag{17}$$

where $n$ represents the total number of peaks in the 3D image; $(x_i, y_i)$ represents the central coordinates of the $i$th peak; $h_i$ represents the height of the highest point of the $i$th peak; $x_{si}$ and $y_{si}$ represent the decay coefficients of the $i$th peak along the $x$ and $y$ axes respectively, which control the slope of the peak.

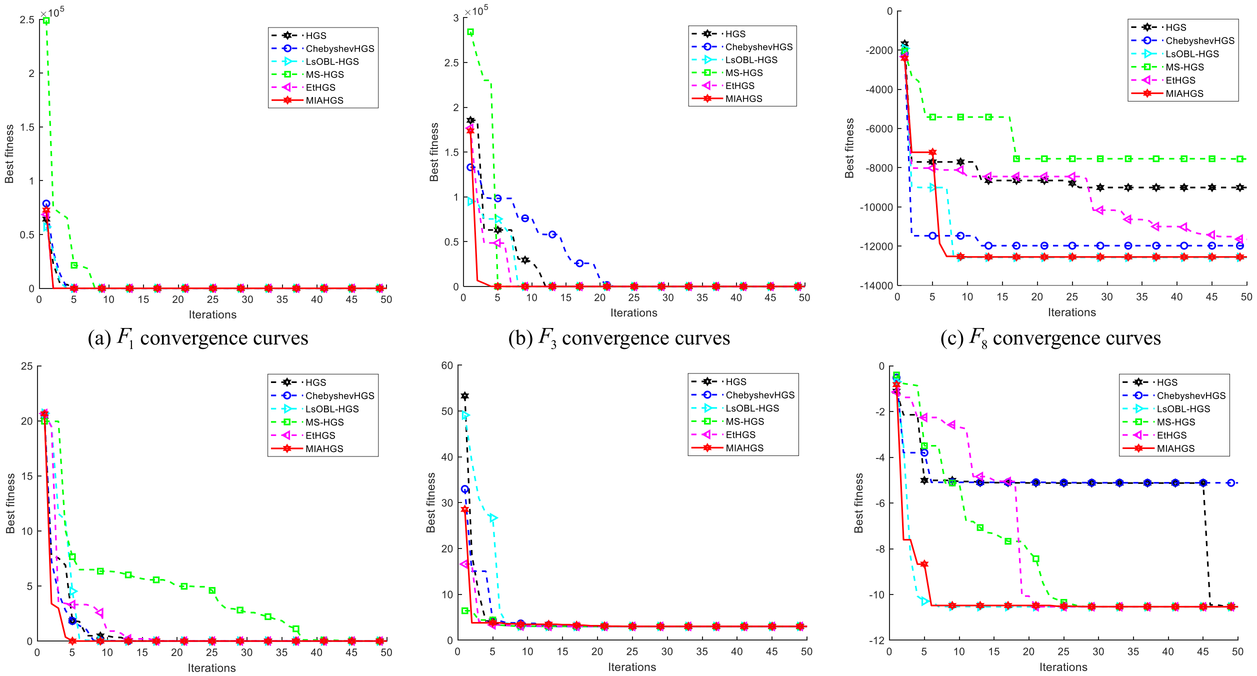
### B. PATH MODEL

In 3D spatial path planning, to avoid obstacles the UAV flight trajectory must be a 3D spatial curve, i.e. with continuous and no abrupt changes in curvature and deflection. However, in the swarm intelligence optimization algorithm, multiple sample points can only be used instead of a curve for optimization. In order to realistically simulate the UAV trajectory, this paper uses cubic B-spline interpolation to smoothly connect the discrete points. $k$ discrete sample points are set as intermediate path nodes, i.e. $\{M_1(x_1, y_1, z_1), \cdots, M_k(x_k, y_k, z_k)\}$; the start and end points are $S(x_s, y_s, z_s)$ and $G(x_g, y_g, z_g)$ respectively, thus forming the complete path node $\{S, M_1, \cdots, M_k, G\}$. These discrete points' coordinates $(x_s, x_1, \cdots, x_k, x_g)$, $(y_s, y_1, \cdots, y_k, y_g)$, and $(z_s, z_1, \cdots, z_k, z_g)$ are fitted using cubic B-spline interpolation, resulting in a smooth 3D spatial curve.

In the path planning process, in order to prevent collisions between the UAV and the mountain, the 3D spatial curve of the simulated route cannot have any crossover overlap with the mountain, meaning that the height $z$ of the curve at the same horizontal and vertical coordinates is greater than the height of the peak $z_h$, i.e.: $z > z_h(x, y)$.

### C. EXPERIMENTAL RESULTS AND ANALYSIS

In order to verify the effectiveness of the MIA-HGS algorithm in 3D spatial path planning, simulation experiments were conducted using Matlab2018a and compared with some classical optimization algorithm. The experimental space is



(a) $F_1$ convergence curves

(b) $F_3$ convergence curves

(c) $F_8$ convergence curves

(d) $F_{10}$ convergence curves

(e) $F_{18}$ convergence curves

(f) $F_{23}$ convergence curves

**FIGURE 4.** **Comparison chart of convergence of 6 improved algorithms.**

**TABLE 4.** **6 mountain parameters.**

| Peak ID | $(x_i, y_i)$ | $h_i$ | $x_{si}$ | $y_{si}$ |
|---|---|---|---|---|
| 1 | (20,10) | 20 | 5.5 | 5 |
| 2 | (40,25) | 35 | 8 | 7 |
| 3 | (45,50) | 25 | 5 | 6 |
| 4 | (60,30) | 38 | 4.5 | 5.5 |
| 5 | (20,45) | 20 | 5.5 | 6 |
| 6 | (30,10) | 25 | 3.5 | 4.5 |

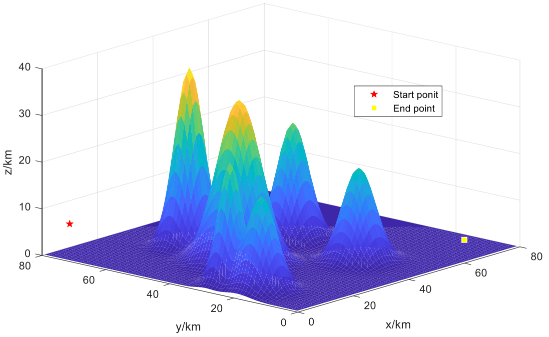


**FIGURE 5.** **Comparison chart of convergence of 6 improved algorithm.**

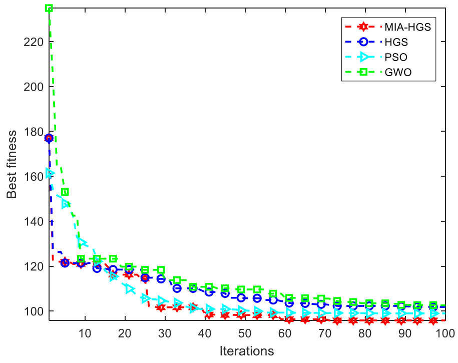


**FIGURE 6.** **Comparison of the convergence of the 4 algorithms on 3D path planning.**

$80 \times 80 \times 40(km^3)$ and there are 6 peaks in the space, the specific experimental data of each peak are shown in Tab. 4. The coordinates of the start and end points are (10, 80, 5) and (60, 0, 5) respectively. The three-dimensional spatial topography is shown in Fig. 5.

The initial population size of the algorithms involved in the experiments was all set to 30, the maximum number of iterations was 100, and the rest of the parameters were kept consistent with the experiments above. The convergence curves of the algorithm iterations are shown in Fig. 6, and the results are shown in Tab. 5.

From Fig. 6 and Tab. 5, it can be seen that MIA-HGS also has a strong optimization capability when solving 3D route planning, and the final shortest route distance obtained by MIA-HGS is $96.71km$, which is better than the $100.93km$ planned by the HGS algorithm, which shows that the HGS and GWO algorithms fall into a local optimum, while PSO is close to the global optimum but converges too slowly. The average of the MIA-HGS algorithm was found to be better

**TABLE 5.** Performance comparison of ELHGS and HGS algorithms.

| Algorithms | Length /km | Average time /s |
|---|---|---|
| MIA-HGS | 96.71 | 0.74 |
| HGS | 100.93 | 0.567 |
| PSO | 98.75 | 0.737 |
| GWO | 102.53 | 0.583 |

than the HGS algorithm. MIA-HGS takes a similar amount of time to HGS in the initialization and adaptive strategy phases, but the final step is more time-consuming. The jump out of the local optimum phase is the generation of a new population based on the original population, which is eventually retained on merit, and consumes similar time as the initialization population with a time complexity of $O(n)$. However, this phase also occurs conditionally and is not executed every time. Overall the overall time complexity of MIA-HGS is increased, but as this phase allows the algorithm to find the global optimum quickly, the extra time spent iterating is more than worth it. The final route map generated by MIA-HGS is shown in Fig. 7, where the route completely bypasses terrain such as peaks and the curve is smooth and continuous. The simulation results show that the MIA-HGS algorithm has successfully solved the 3D path planning problem.

The fitness parameters in MIA-HGS remain key to controlling whether populations choose global search or local exploration. The introduction of variation is at an intermediate to late stage and variation manipulation does not affect the choice between global search and local exploration.

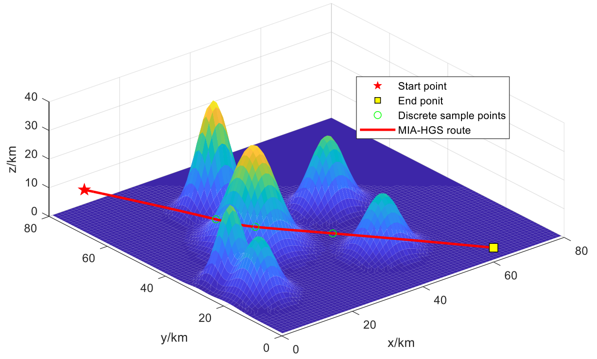


**FIGURE 7.** MIA-HGS planned 3D route map.

## VI. CONCLUSION AND FUTURE PERSPECTIVES

This study proposes a Multi-strategy fusion Improved Adaptive Hunger Games Search algorithm (MIA-HGS) to address the disadvantages of poor initial population diversity, low solution accuracy, and the tendency to fall into local optimality of the Hunger Game Search algorithm (HGS). The algorithm uses a good point set strategy to enhance the diversity of the initial population and provide a good basis for the algorithm to find an optimum; The original HGS algorithm uses a fixed parameter to control the selection of the update strategy during the convergence process. In this paper, the parameter is improved by replacing the fixed parameter with an adaptive adjustment parameter, which automatically adjusts the parameter size according to the number of iterations and selects the update strategy according to the situation, further improving the global search ability and reducing the probability of the population falling into a local optimum; when the population stagnates, a logarithmic spiral opposition-based learning strategy is introduced to enable the algorithm to jump out of the local optimum in time and accelerate the speed of the search. Adaptive parameters in MIA-HGS are still key to controlling the choice of global search or local exploration of populations. The introduction of variation is at an intermediate to late stage and the variation operation does not influence the choice between global search and local exploration. These improvements have led to significant improvements in the accuracy, stability, and speed of convergence of the algorithm.

Finally, through simulation experiments on 23 benchmark test functions and comparison with five traditional optimization algorithms and four improved HGS algorithms, the experimental results show that the MIA-HGS algorithm effectively improves the ability of the original algorithm to search globally and jump out of local optima in high-dimensional problems, improves the convergence accuracy and accelerates the convergence speed. In order to verify the ability of MIA-HGS in solving practical problems, this paper applies MIA-HGS to UAV trajectory planning and also achieves good application results, and accomplishes the planning objectives better. The next step will be to use the MIA-HGS algorithm on more practical engineering problems, such as friction welding, image segmentation, bridge design, etc., to verify the algorithm's ability to solve more and more complex practical problems.

## REFERENCES

[1] G. K. Wen, Y. Dasril, and A. R. Othman, "A class of ABCED conjugate gradient method in solving general global optimization problems," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 23, pp. 7984–7995, Dec. 2018.

[2] X. Liu, P. Zhang, H. Fang, and Y. Zhou, "Multi-objective reactive power optimization based on improved particle swarm optimization with e-greedy strategy and Pareto archive algorithm," *IEEE Access*, vol. 9, pp. 65650–65659, 2021, doi: 10.1109/ACCESS.2021.3075777.

[3] H. Tseng, C. Chang, and T. Chung, "Applying improved particle swarm optimization to asynchronous parallel disassembly planning," *IEEE Access*, vol. 10, pp. 80555–80564, 2022, doi: 10.1109/ACCESS.2022.3195863.

[4] A. K. V. K. Reddy and K. V. L. Narayana, "Symbiotic learning grey wolf optimizer for engineering and power flow optimization problems," *IEEE Access*, vol. 10, pp. 95229–95280, 2022, doi: 10.1109/ACCESS.2022.3203999.

[5] C. Li, F. You, T. Yao, J. Wang, W. Shi, J. Peng, and S. He, "Simulated annealing particle swarm optimization for high-efficiency power amplifier design," *IEEE Trans. Microw. Theory Techn.*, vol. 69, no. 5, pp. 2494–2505, May 2021.

[6] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1109/TMTT.2021.3061547.

[7] J. Xue, "Research and application of a novel swarm intelligence optimization technique: Sparrow search algorithm," M.A thesis, College Inf. Sci. Technol., Donghua Univ., Shanghai, China, Dec. 2019, doi: 10.27012/d.cnki.gdhuu.2020.000178.

[8] H. M. Hasanien, A. A. El-Fergany, and A. M. Agwa, "Semi-empirical PEM fuel cells model using whale optimization algorithm," *Energ. Convers. Manag.*, vol. 201, Nov. 2019, Art. no. 112197, doi: 10.1016/j.enconman.2019.112197.

[9] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Exp. Syst. Appl.*, vol. 177, Sep. 2021, Art. no. 114864, doi: 10.1016/j.eswa.2021.114864.

[10] Y. Yang, Y. Wu, H. Yuan, M. Khishe, and M. Mohammadi, "Nodes clustering and multi-hop routing protocol optimization using hybrid chimp optimization and hunger games search algorithms for sustainable energy efficient underwater wireless sensor networks," *Sustain. Comput., Informat. Syst.*, vol. 35, Sep. 2022, Art. no. 100731.

[11] W. S. AbuShanab, M. A. Elaziz, E. I. Ghandourah, E. B. Moustafa, and A. H. Elsheikh, "A new fine-tuned random vector functional link model using hunger games search optimizer for modeling friction stir welding process of polymeric materials," *J. Mater. Res. Technol.*, vol. 14, pp. 1482–1493, Sep. 2021.

[12] S. Yu, A. A. Heidari, C. He, Z. Cai, M. M. Althobaiti, R. F. Mansour, G. Liang, and H. Chen, "Parameter estimation of static solar photovoltaic models using Laplacian Nelder–Mead hunger games search," *Sol. Energy*, vol. 242, pp. 79–104, Aug. 2022.

[13] C. Xie, H. Nguyen, Y. Choi, and D. J. Armaghani, "Optimized functional linked neural network for predicting diaphragm wall deflection induced by braced excavations in clays," *Geosci. Frontiers*, vol. 13, no. 2, Mar. 2022, Art. no. 101313.

[14] M. Togaçar, "Detecting attacks on IoT devices with probabilistic Bayesian neural networks and hunger games search optimization approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 1, p. e4418, Dec. 2021.

[15] L. Hou, R. Li, M. Mafarja, A. A. Heidari, L. Liu, C. Jin, S. Zhou, H. Chen, Z. Cai, and C. Li, "Image segmentation of intracerebral hemorrhage patients based on enhanced hunger games search optimizer," *Biomed. Signal Process. Control*, vol. 82, Apr. 2023, Art. no. 104511.

[16] Y. O. Shaker, D. Yousri, A. Osama, A. Al-Gindy, E. Tag-Eldin, and D. Allam, "Optimal charging/discharging decision of energy storage community in grid-connected microgrid using multi-objective hunger game search optimizer," *IEEE Access*, vol. 9, pp. 120774–120794, 2021, doi: 10.1109/ACCESS.2021.3101839.

[17] F. Kutlu Onay and S. B. Aydemir, "Chaotic hunger games search optimization algorithm for global optimization and engineering problems," *Math. Comput. Simul.*, vol. 192, pp. 514–536, Feb. 2022.

[18] B. J. Ma, S. Liu, and A. A. Heidari, "Multi-strategy ensemble binary hunger games search for feature selection," *Knowl.-Based Syst.*, vol. 248, Jul. 2022, Art. no. 108787.

[19] S. Chakraborty, A. K. Saha, R. Chakraborty, M. Saha, and S. Nama, "HSWOA: An ensemble of hunger games search and whale optimization algorithm for global optimization," *Int. J. Intell. Syst.*, vol. 37, no. 1, pp. 52–104, Sep. 2021.

[20] D. Izci, S. Ekinci, E. Eker, and M. Kayri, "Augmented hunger games search algorithm using logarithmic spiral opposition-based learning for function optimization and controller design," *J. King Saud Univ.-Eng. Sci.*, Mar. 2022.

[21] Z. J. Li and J. X. Cheng, "Uniform design of initial population of genetic algorithm based on good point set," *Comput. Inf. Technol.*, vol. 4, pp. 29–32, Aug. 2007.

[22] Y. F. Xu, S. Sheng, W. K. Zhang, and Y. S. Liu. *Elite Opposition-Based Learningandt-Distribution Hunger Games Search Algorithm*. Accessed: Aug. 2022. [Online]. Available: http://kns.cnki.net/kcms/detail/11.3724.TP.20220824.1444.002.html

[23] Y. Wang, *Applications of Number Theory in Approximate Analysis*. Beijing, China: Science Press, 1978, pp. 83–86.

[24] Y. Chai, G. Li, S. Ren, and Z. Xu, "Multi-strategy fusion improved adaptive tunicate swarm algorithm," *Appl. Res. Comput.*, Apr. 2023.

[25] G. He and X.-L. Lu, "Good point set and double attractors based-QPSO and application in portfolio with transaction fee and financing cost," *Exp. Syst. Appl.*, vol. 209, Dec. 2022, Art. no. 118339.

[26] Z. Feng, J. Li, X. Ge, and Y. Zhou, "Integrating multi strategy improved whale optimization algorithm and its application," *Comput. Integr. Manuf. Syst.*, Jan. 2023. [Online]. Available: http://kns.cnki.net/kcms/detail/11.5946.tp.20230104.1215.014.html

[27] H. R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence," in *Proc. Int. Conf. Comput. Intell. Modelling, Control Automat., Int. Conf. Intell. Agents, Web Technol. Internet Commerce*, vol. 1, Nov. 2005, pp. 695–701.

[28] S. Garcia, A. Fernandez, J. Luengo, and F. Herrera, "Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Exp. Anal. Power, Inf. Sci.*, vol. 180, pp. 2044–2064, May 2021, doi: 10.1016/j.ins.2009.12.010.

[29] D. Zhang, J. Xu, Y. Zhao, and J. Ding, "Bidirectional search grey wolf optimizer based on stagnation detection," *Appl. Res. Comput.*, vol. 39, no. 6, pp. 1725–1730, Jan. 2022.

[30] H. Shuzhao, T. Junwei, and Q. Lu, "Unmanned aerial vehicle path planning based on improved genetic algorithm," *J. Comput. Appl.*, vol. 39, no. 2, pp. 390–397, Feb. 2021.

**DAMING ZHANG** received the Ph.D. degree in resource information and decision-making from Northeastern University, China, in 2017. He is currently an Associate Professor and a Master's Supervisor with the School of Information Science and Engineering, Guilin University of Technology. His main research interests include resource information, decision-making, and software engineering.

**YANQING ZHAO** is currently pursuing the master's degree with the School of Information Science and Engineering, Guilin University of Technology, Guilin, China.

**JUNJIE DING** is currently pursuing the master's degree with the School of Information Science and Engineering, Guilin University of Technology, Guilin, China.

**ZIJIAN WANG** is currently pursuing the master's degree with the School of Information Science and Engineering, Guilin University of Technology, Guilin, China.

**JIAQING XU** is currently pursuing the master's degree with the School of Information Science and Engineering, Guilin University of Technology, Guilin, China.