HYBRID GRASSHOPPER OPTIMIZATION ALGORITHM INCORPORATING WHALE OPTIMIZATION ALGORITHM

Wei LIU[1,*], Guangyu HAN[2], Tong LI[3], Tengteng REN[4], Wenlv YAN[5]

In order to address the issues with the original grasshopper optimization algorithm, this work offers a hybrid grasshopper optimization algorithm (WOGOA). First, the initial population of grasshoppers was mapped by Logistic mapping. Second, the parameter c was changed, and the nonlinear weight was added. Finally, Levy flight was integrated into the spiral bubble net hunting behavior of the whale optimization algorithm and then introduced into the grasshopper optimization algorithm as a whole. The algorithm was benchmarked on 9 test functions. According to the experimental data, WOGOA can deliver outcomes that are highly competitive in terms of convergence ability and accuracy.

Keywords: Grasshopper optimization algorithm; Whale optimization algorithm; Logistic chaotic maps; Levy fight

1. Introduction

Fusion algorithm refers to the combination of certain behaviors of two or more algorithms to complement each other's strengths and weaknesses, so as to better achieve the effect before fusion. For example, reference [1] proposed a new improved whale optimization algorithm (IWOA) of multi-strategy hybrid algorithm, which enhanced the speed of convergence and global search capabilities of whale optimization algorithm. The various performances of the algorithm were benchmarked on 23 benchmark functions, proving the superiority of IWOA. In reference [2], the author demonstrated the use of modified whale optimization algorithm (MWOA) search based Selective Harmonic Elimination Pulse Width Modulation application, which has contributed to the field of power quality in micro grid systems. Reference [3] proposed a new multi-objective particle swarm

[1] Prof., College of Information Science and Engineering, Shenyang LIGONG University, China, e-mail: 251306474@qq.com
[2] Pg., College of Information Science and Engineering, Shenyang LIGONG University, China, e-mail: hanguangyu258@163.com
[3] College of Information Science and Engineering, Shenyang LIGONG University, China; 3536360384@qq.com
[4] College of Information Science and Engineering, Shenyang LIGONG University, China; 2997433958@qq.com
[5] College of Information Science and Engineering, Shenyang LIGONG University, China; yanwenlv027@163.com

optimization algorithm and used it to optimize the geometric shape of permanent magnet motors and compared it with two other multi-objective optimization particle swarm algorithms, demonstrating its superiority. Through this literature, we can comprehend that the fusion algorithm performs well in both its own right and other disciplines, demonstrating its effectiveness.

The whale optimization algorithm (WOA) [4] and grasshopper optimization algorithm (GOA) [5] both imitate the predatory tendencies of their respective animal counterparts in the wild. The principle of GOA is easy to understand, the parameters are few, and it is easy to implement. However, the algorithm convergence speed is slow and prone to premature convergence. Improved in response to the limitations of GOA, this article enhances its effectiveness. Due to the initial population distribution of GOA is random, the richness and uniformity of the population are reduced. Therefore, this article uses Logistic chaotic mapping to initiate the grasshopper population. Additionally, the GOA parameter c has a significant impact on the algorithm's capacity for exploration and exploitation. However, a linear decrease in c does not promote optimal balance within the algorithm. Therefore, the linearly reduced parameter c is changed to a non-linear reduction. WOA only updates its position based on the best individual, which results in a lack of population diversity and low global search ability. Furthermore, numerous academics have demonstrated that Levy flight can depart from local optima. Add the above methods to GOA to obtain WOGOA. The data results of WOGOA on the test function prove that WOGOA has high optimization ability.

2. Fundamental principles of algorithm

2.1. Grasshopper Optimization Algorithm

It imitates the swarming and foraging behaviors of grasshoppers in the wild. The goal of algorithmic biomimetic is to quickly translate the small-scale movement behavior of grasshopper larvae to local development. The large-scale mobility behavior of adult grasshoppers is mapped to a long-term global search. The algorithm's optimization phase is the process of looking for food sources. Equation (1) can be used to explain grasshopper swarm behavior.

$$X_i = c\left(\sum_{i=1, i\neq j}^{N} c\frac{ub-lb}{2} s(|x_j - x_i|)\frac{x_j - x_i}{d_{ij}}\right) + \hat{T}_d , \tag{1}$$

where X_i is the position of grasshopper; *ub* and *lb* are the upper and lower bounds of the search space, respectively; x_j and x_i are different grasshoppers; d_{ij} is the distance between different x_j and x_i grasshoppers; $\widehat{T_d}$ is the optimal location. *s* is calculated by formula (2), where *f*=0.5 and *l*=1.5.

$$s(r) = fe^{\frac{-r}{l}} - e^{-r} \tag{2}$$

The parameter outside the parentheses c of equation (1) can balance the entire grasshopper swarm's exploration and exploitation. The purpose of the parameter c inside the parentheses is to limit the range of the repulsion force between grasshoppers. It is calculated as equation (3):

$$c = c_{max} - t\,\frac{c_{max} - c_{min}}{T_{max}}, \tag{3}$$

where t denotes the current iteration count and T_{max} denotes the maximum number of iterations; $c_{\max}$ is equal to 1 and $c_{\min}$ is equal to 0.0004.

2.2. Bubble net hunting behavior in Whale Optimization Algorithm

Humpback whales engage in one of two bubble-net behaviors: shrinking encircle and spiral updating position. The bubble net behavior is shown in Fig. 1. The spiral update position is shown in equation (4):

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t), \tag{4}$$

where the helix's shape can be altered by the constant b, which has a value of 1; $\vec{X}(t+1)$ is the position of the next iteration; $\vec{X}^*(t)$ is the position vector of the optimal solution; l is a selection value between [-1, 1]; $\vec{D}'$ can be calculated from equation (5) to obtain.

$$\vec{D}' = \left|\vec{X}^*(t) - \vec{X}(t)\right|, \tag{5}$$

$\vec{X}(t)$ is the position vector.

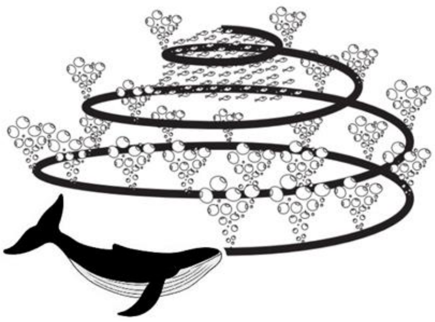


Fig. 1. A humpback whale's bubble-net eating technique

3. WOGOA

3.1. Chaotic Map Initialization

In the area of algorithm optimization, chaotic mapping can frequently substitute the pseudorandom generator for generating chaotic numbers from 0 to 1. Reference [6] applied the Tent mapping to the grasshopper optimization algorithm and achieved good results. The Logistic map utilized in this article is widely adopted due to its straightforward expression, as shown in equation (6):

$$X_{n+1} = \lambda \times X_n \times (1 - X_n). \tag{6}$$

In the formula, X_{n+1} in [0, 1], λ in [0, 4] is a parameter of Logistic, which has different chaotic effects in different values. Reference [7] indicates that when 3.5669<λ⩽ 4. This mapping presents a completely chaotic state, so this article chooses λ = 4. A random number with an initial position between 0 and 1.

3.2. Nonlinear parameters

From equation (1), it follows that parameter *c* is crucial for achieving a balance between exploration and exploitation during the GOA algorithm's process of updating the grasshopper position. However, the linear variation of parameter *c* hinders effective global and local search, leading to low convergence accuracy [8]. This paper suggests a novel nonlinear parameter to address this issue, as shown in equation (7):

$$c = (c_{max} - \frac{t}{T_{max}})^3, \tag{7}$$

c_{max}, c_{min}, and T_{max} have been elaborated on in the previous text. The rapid decrease of parameter *c* in the early phase prevents the algorithm from becoming stagnated into local optimum, while its slower decrease in later phases allows for more detailed exploitation. Therefore, this nonlinear parameter can better balance the algorithm's exploration and exploitation.

3.3. Nonlinear Weights

From the preceding description, the grasshopper position is updated using equation (1) throughout the whole iteration phase. However, in the later phase, this equation can only promote the grasshopper to approach the target position, but cannot converge to the global optimum faster [9]. Being inspired by the particle swarm weights mentioned in reference, nonlinear weights were added to the GOA [10]. The nonlinear weight w is shown in equation (8):

$$w = w_{max} - (w_{max} - w_{min}) \times (\frac{t}{T_{max}})^{\frac{1}{t^{y^2}}}, \tag{8}$$

where the current iteration is *t*; w_{max} is 1, w_{min} is 0.0004; the maximum iteration is T_{max}; *y* can be calculated by equation (9):

$$y = \exp(-\log_{10}^{t/10}). \tag{9}$$

From equation (8), it is evident that the algorithm's exploration ability is enhanced during the early search phase due to a relatively large weight, ensuring global search. As the algorithm progresses, the weight gradually decreases, improving its exploitation performance and convergence speed. Therefore, the position update of grasshoppers becomes equation (10):

$$X_i = c(\sum_{i=1, i\neq j}^{N} c\frac{ub - lb}{2} s(|x_j - x_i|)\frac{x_j - x_i}{d_{ij}}) + w \times T_d. \tag{10}$$

3.4. Levy flight was integrated into the spiral bubble-net attacking

The inclusion of the Levy flight mechanism improves the algorithm's ability to conduct global exploration while looking for the best values, making it easier to discover the global optimum. The Levy flight follows the Levy distribution, where the formula is shown in equation (11):

$$X_i^{t+1} = X_i^t + \alpha \oplus \text{Levy}\,, \tag{11}$$

where X_i^{t+1} represents the next position, X_i^t represents the current iteration position, and step size α is usually a constant with a value of 0.01. However, the method in this article can have a big step size for global search the algorithm in its early phases, making it simple to jump out of local optima. So the value of an in this article ranges between [0.01, 0.3]. $\oplus$ representing point-to-point multiplication. And the Levy is described by equation (12):

$$Levy = (s_1, s_2, \cdots, s_n),\, s_i = \frac{\mu}{|v|^{\frac{1}{\lambda}}},\, i = 1, 2, \cdots, n\,, \tag{12}$$

where v adheres to the normal distribution with a 0 mean and a 1 variance, μ obey mean of 0, variance of σ_μ the Gaussian distribution, where σ_μ the calculation method is shown in equation (13):

$$\sigma_\mu = \left[\frac{\Gamma(1+\lambda)\times\sin\left(\pi\times\frac{\lambda}{2}\right)}{\Gamma\left(\frac{1+\lambda}{2}\right)\times\lambda\times 2^{\frac{\lambda-1}{2}}}\right]^{\frac{1}{\lambda}}, \tag{13}$$

λ is a random number in (1, 3), in this article we assumed it as 1.5; $\Gamma(x)$ is the gamma function.

After integrating Levy flight into the spiral bubble-net attacking behavior of whales, the updated formula becomes:

$$\vec{X}(t+1) = \vec{X}_\alpha(t) + Levy \cdot \left|\vec{X}_\alpha - \vec{X}(t)\right| \cdot e^{bl} \cdot \cos(2\pi l)\,, \tag{14}$$

where b is the constant coefficient of the spiral equation, usually b is 1; l is a selection value between [-1, 1].

Integrating Levy flights into the spiral bubble-net attacking behavior facilitates running out of local optima to search global optima during the optimization process. Combining the position update of grasshoppers with the position update of whale spiral bubble-net attacking behavior which was added to Levy flight, and alternate between them for updating. Specifically, execute equation (10) when the number of iterations is odd and equation (14) when it is even.

3.4. WOGOA Step

Step 1 Make the population's initial state via logistic mapping, set parameters c_{max}, c_{min}, population N, and the most iterations allowed *Maxiter*.

Step 2 Determine the starting population's fitness, and find the best T_d position for the current population after updating the target position.

Step 3 When the current iteration count is singular, execute the position update equation (10); otherwise, execute the position update equation (14).

Step 4 Edge the new position *X'(t)* obtained, calculate the fitness, and compare it with position *X(t)* in the previous iteration. Find the ideal position, update T_d, and save the position with the lowest fitness as the final position *X(t+1)*.

Step 5 Ascertain if the reached number of iterations has been reached. When the maximum number of iterations has been achieved, publish the overall best solution; if not, continue with steps 3 and 4.

In Fig. 2, the WOGOA flowchart is displayed.

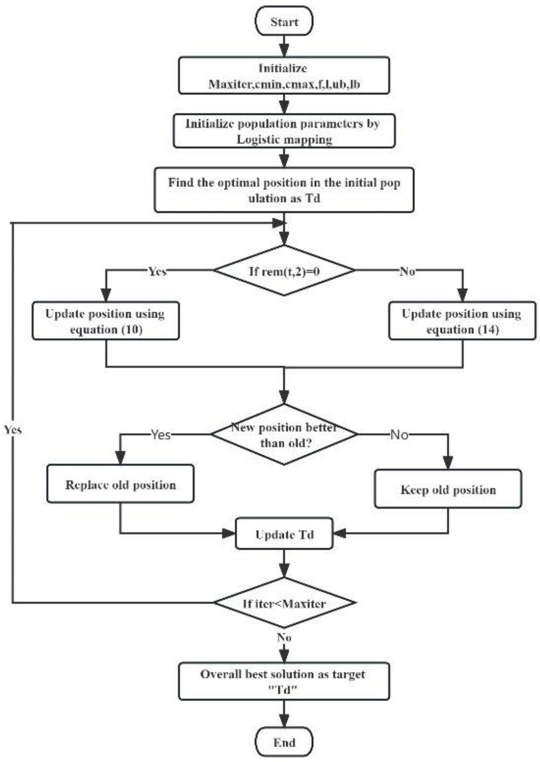


Fig. 2. The flowchart of WOGOA

4. Results

4.1. Experimental environment

The experiments are tested on an Intel machine Core(TM) i5-8250U CPU 1.6GHz and 20 GB RAM. All algorithms are tested using the MATLAB 2022b software.

4.2. Comparison with different original algorithms

To prove the WOGOA algorithm's capacity for optimization, this paper selected Ant lion optimization algorithm (ALO) [11], Harris hawks optimization (HHO) [12], Whale optimization algorithm (WOA), and Grasshopper optimization algorithm (GOA) to compare with WOGOA. The number of search agents is 30, and other parameters were the same as those in the reference. This paper selected 9 test functions for comparison. In Table 1, which corresponds to rows F1 through F9 in the test data result table, the names, dimensions, and formulas of the chosen functions are displayed. F1 through F6 are unimodal functions having a single best solution. The proposed algorithm's rate of convergence will be examined. F7 to F9 are multimodal functions with numerous global solutions. The purpose is to verify WOGOA's global optimization ability and whether it can find the global optimal solution. Each test function was run 30 times independently to avoid the occasionality of a single experiment, and the number of iterations was set to 1000.

Table 1

Partial benchmark functions

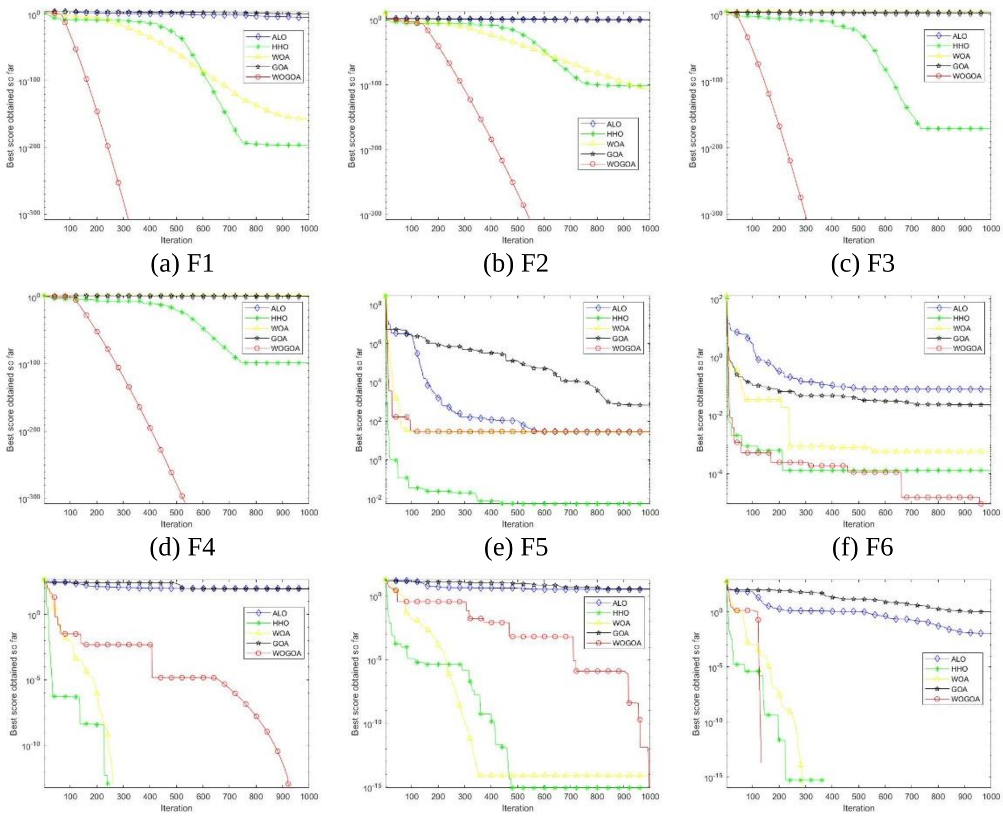
Function name	Function	Range	dim	$f_{\min}$
Sphere (F1)	$f(x)=\sum_{i=1}^{\dim} x_i^2$	[-100,100]	30	0
Schwefel 2.22 (F2)	$f(x)=\sum_{i=1}^{\dim}\lvert x_i\rvert+\prod_{i=1}^{\dim}\lvert x_i\rvert$	[-10,10]	30	0
Schwefel 1.2 (F3)	$f(x)=\sum_{i=1}^{\dim}(\sum_{j=1}^{i} x_j{}^2)$	[-100,100]	30	0
Schwefel 2.21 (F4)	$f(x)=\max_i\lvert x_i\rvert,1\le i\le\dim$	[-100,100]	30	0
Rosenbrock (F5)	$f(x)=\sum_{i=1}^{\dim-1}\left[100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2\right]$	[-30,30]	30	0
Quartic (F6)	$f(x)\sum_{i=1}^{\dim} ix_i^4+random[0,1)$	[-1.28,1.28]	30	0
Rastrigin (F7)	$f(x)=\sum_{i=1}^{\dim}[x_i^2-10\cos(2\pi x_i)+10]$	[-5.12,5.12]	30	0
Ackley (F8)	$f(x)=-20\exp(-0.2\sqrt{\frac{1}{\dim}\sum_{i=1}^{\dim}x_i^2})-\exp(\frac{1}{\dim}\sum_{i=1}^{\dim}\cos(2\pi x_i))+20+e$	[-32,32]	30	0
Griewank (F9)	$f(x)=\frac{1}{4000}\sum_{i=1}^{\dim}x_i^2-\prod_{i=1}^{\dim}\cos(\frac{x_i}{\sqrt{i}})+1$	[-600,600]	30	0

Table 2 displays the test results data. In Table 2, Best is the optimal value of 30 experiments, Worst is the worst value of 30 experiments, Ave is the average value of 30 experiments, and Std is the variance of 30 experiments. Fig. 3 displays the test's convergence curves.

Table 2

Partial benchmark function results of different algorithms

Function	Algorithms	Best	Worst	Ave	Std
F1	ALO	1.81E-06	1.55E-05	6.33E-06	3.96E-06
	HHO	4.21E-207	8.11E-180	2.70E-181	0
	WOA	5.30E-168	4.58E-150	1.58E-164	8.35E-151
	GOA	0.5741	18.8231	7.3336	5.4888
	WOGOA	0	0	0	0
F2	ALO	0.0890	124.1006	37.6341	51.7361
	HHO	1.10E-111	9.01E-97	7.02E-98	2.12E-97
	WOA	5.44E-113	2.90E-101	1.05E-102	5.30E-102
	GOA	0.2035	71.2622	8.3375	13.7452
	WOGOA	0	0	0	0
F3	ALO	580.8901	3605.0574	1136.3067	601.6023
	HHO	1.80E-184	6.27E-138	2.10E-139	1.14E-138
	WOA	2070.0063	31620.2934	18003.2910	9046.1781
	GOA	790.9804	5217.8573	1983.0377	1214.4968
	WOGOA	0	0	0	0
F4	ALO	4.8288	20.1975	12.8514	4.0509
	HHO	4.77E-105	1.27E-89	4.23E-91	2.31E-90
	WOA	0.0420	85.1367	43.9337	31.2751
	GOA	5.3958	17.3842	9.6231	3.0170
	WOGOA	0	1.69E-20	4.43E-21	6.29E-21
F5	ALO	22.0201	1740.5569	272.5554	475.0298
	HHO	3.47E-06	0.0219	0.0045	0.0057
	WOA	27.0561	28.7849	28.1150	0.4723
	GOA	122.1291	5561.6334	1031.9546	1268.8228
	WOGOA	28.7071	28.9795	28.8075	0.0697
F6	ALO	0.0387	0.2246	0.1012	0.4217
	HHO	8.89E-06	0.0003	7.04E-05	6.66E-05
	WOA	1.51E-05	0.0172	0.0029	0.0040
	GOA	0.0038	0.0301	0.0175	0.0062
	WOGOA	6.89E-06	0.0003	0.0001	0.0001
F7	ALO	39.7983	190.0363	79.2317	28.8098
	HHO	0	0	0	0
	WOA	0	0	0	0
	GOA	43.8411	158.8225	93.5683	31.4093
	WOGOA	0	0	0	0
F8	GOA	2.5101	6.4516	4.0077	0.9158
	SCGOA	1.6183	4.8464	2.6858	0.8751
	NGOA1	1.81E-08	2.04E-08	1.95E-08	7.41E-10
	LGOA	2.4088	4.4249	3.5474	0.6659
	WOGOA	8.88E-16	8.88E-16	8.88E-16	0
F9	GOA	0.5371	0.9851	0.7473	0.1341
	SCGOA	0.1797	0.3938	0.3011	0.0791
	NGOA1	9.88E-05	1.40E-14	1.26E-14	1.45E-15
	LGOA	0.1076	0.3116	0.2022	0.0703
	WOGOA	0	0	0	0



(a) F1 (b) F2 (c) F3

(d) F4 (e) F5 (f) F6

(g) F7 (h) F8 (i) F9

Fig. 3. Comparison of convergence curves of different algorithms

As shown in Table 2, both WOA and HHO exhibit stronger overall optimization ability than ALO and GOA from F1-F2. Furthermore, based on the table's standard deviation, it can be observed that WOA and HHO exhibit superior robustness compared to ALO and GOA. This suggests that their optimization capabilities are not merely coincidental. However, WOGOA outperforms both WOA and HHO. It has good optimization ability in terms of precision and consistency, enabling it to achieve the theoretical optimal value.

From the perspective of F3-F4, WOA's overall optimization ability is inferior to that of ALO and GOA. Neither method can achieve the theoretical optimal value. And the stability of WOA is not as good as that of ALO and GOA. The overall optimization ability of HHO is much better than WOA, ALO, and GOA, and the standard deviation is low, suggesting that HHO is more stable. After comparing WOGOA with the other four algorithms, it is found that WOGOA exhibits significantly superior global optimization ability, convergence speed, and stability, and is capable of achieving the theoretical optimal value.

From the point of view of F5, WOA and WOGOA have similar optimization abilities. But in terms of stability, WOWGOA is a bit more stable than WOA. But none of these five algorithms can locate the ideal value in theory.

From F6, the optimal values found by WOGOA, HHO, and WOA are superior to the other two algorithms. From the comparison of these three algorithms, WOGOA and HHO are better than WOA in terms of accuracy and stability. WOGOA's optimization ability is slightly stronger than HHO's, but neither of them is able to determine the theoretical optimal value. From the perspective of F7-F9, compared to the other two algorithms, WOA, HHO, and GWGOA can run the local optimal solution and locate the theoretically ideal value at the global level, and the stability is stronger than the other two algorithms. According to the above information obtained from the test data table and convergence curve, GWGOA can find the optimal value on most functions, which proves the effectiveness of GWGOA and highlights the excellent optimization ability of this algorithm.

4.3. Comparison with the improved GOA

WOGOA was contrasted with original GOA, SCGOA [13], LGOA [14], and NGOA1 [15] in order to further illustrate the optimization capability of WOGOA. The settings were set in accordance with the references, with the number of grasshoppers set to 30, the dimension set to 30, the number of iterations set to 1000, and the number of grasshoppers set to 30. Table 3 displays the test data results, and Figure 4 displays the fitness convergence curve.

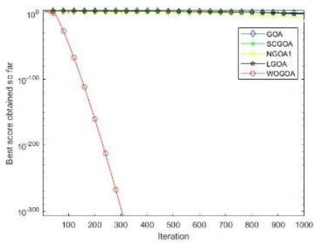
It is evident from Table 3 and Fig. 4 that WOGOA in F1-F4 exhibits superior optimization capabilities compared to the other four algorithms and can effectively identify the optimal solution within the function theory.

Table 3

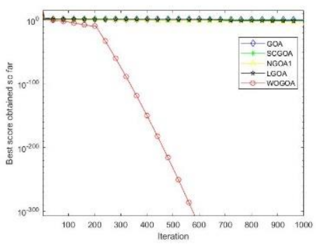
Partial benchmark function results of different improved GOA algorithms

Function	Algorithms	Best	Worst	Ave	Std
F1	GOA	0.5741	18.8231	7.3336	5.4888
	SCGOA	0.2768	0.7125	0.4331	0.1424
	NGOA1	2.89E-17	2.64E-17	2.78E-17	8.32E-17
	LGOA	0.0567	0.9522	0.2725	0.2646
	WOGOA	0	0	0	0
F2	GOA	0.2035	71.2622	8.3375	13.7452
	SCGOA	1.7461	11.8204	4.3410	2.9065
	NGOA1	2.43E-09	2.33E-09	2.40E-09	5.50E-11
	LGOA	0.3203	23.5404	5.2994	7.2088
	WOGOA	0	0	0	0
F3	GOA	790.9804	5217.8573	1983.0377	1214.4968
	SCGOA	397.1571	1154.8540	652.6811	240.2682
	NGOA1	7.48E-15	1.71E-14	1.45E-14	2.85E-15
	LGOA	85.4954	292.0814	158.1364	70.5805
	WOGOA	0	0	0	0
	GOA	5.3958	17.3842	9.6231	3.0170

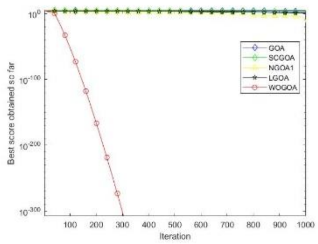
F4	SCGOA	2.8756	20.9253	8.1670	5.2974
	NGOA1	2.63E-05	2.73E-08	2.69E-08	3.58E-10
	LGOA	4.6534	27.7915	15.1358	6.6521
	WOGOA	0	1.69E-20	4.43E-21	6.29E-21
F5	GOA	122.1291	5561.6334	1031.9546	1268.8228
	SCGOA	55.5941	274.8262	136.0421	63.7939
	NGOA1	28.9352	28.9830	28.9552	0.0144
	LGOA	28.2851	793.2353	160.5964	228.3359
	WOGOA	28.7071	28.9795	28.9960	0.0697
F6	GOA	0.0038	0.0301	0.0175	0.0062
	SCGOA	0.0783	0.3871	0.2404	0.0875
	NGOA1	1.10E-05	0.0002	7.82E-05	7.13E-05
	LGOA	0.0074	0.0315	0.0198	0.0071
	WOGOA	6.89E-06	0.0003	0.0001	0.0001
F7	GOA	43.8411	158.8225	93.5683	31.4093
	SCGOA	72.3972	175.5481	137.1621	33.53442
	NGOA1	0	0	0	0
	LGOA	50.7467	190.0385	91.1780	36.8202
	WOGOA	0	0	0	0
F8	GOA	2.5101	6.4516	4.0077	0.9158
	SCGOA	1.6183	4.8464	2.6858	0.8751
	NGOA1	1.81E-08	2.04E-08	1.95E-08	7.41E-10
	LGOA	2.4088	4.4249	3.5474	0.6659
	WOGOA	8.88E-16	8.88E-16	8.88E-16	0
F9	GOA	0.5371	0.9851	0.7473	0.1341
	SCGOA	0.1797	0.3938	0.3011	0.0791
	NGOA1	9.88E-05	1.40E-14	1.26E-14	1.45E-15
	LGOA	0.1076	0.3116	0.2022	0.0703
	WOGOA	0	0	0	0



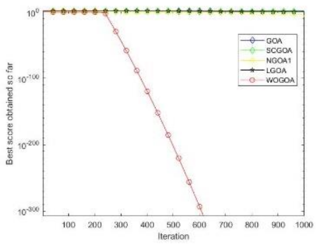
(a) F1



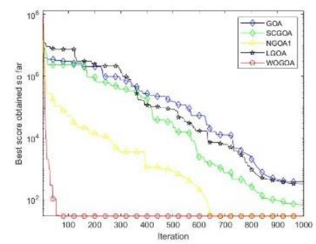
(b) F2



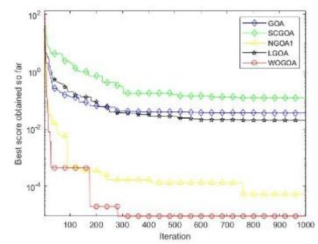
(c) F3



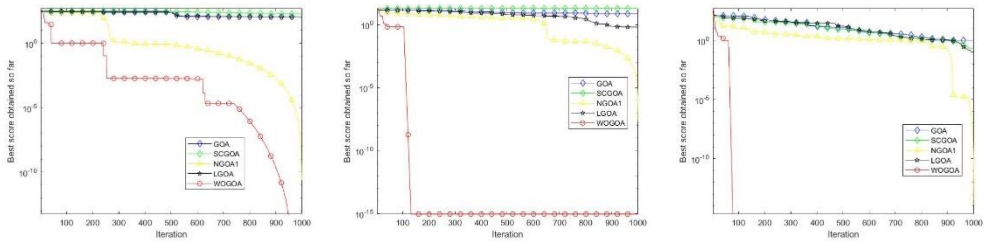
(d) F4



(e) F5



(f) F6



(g) F7 (h) F8 (i) F9

Fig. 4. Comparison of convergence curves of different improved GOA algorithms

Secondly, in terms of standard deviation, WOGOA exhibits the smallest value, indicating a more stable optimization ability and stronger robustness compared to NGOA1. Finally, the rate of convergence of WOGOA is quicker than that of the other four algorithms, indicating that the algorithm has a stronger convergence ability. In general, the convergence speed, accuracy, and stability of WOGOA are higher than those of the other four algorithms in these four functions, which indicates that WOGOA has better optimization ability.

From the perspective of F5, WOGOA and NGOA1 have the same optimization ability on this function, which is better to other algorithms in terms of optimization precision and stability. The theoretically ideal value of this function cannot be found by any of the five algorithms mentioned above.

According to F6, GOA, SCGOA, and LGOA all have the same level of optimization capability in terms of accuracy and stability, and all of them have good accuracy and robustness. WOGOA still has a lot of room for improvement since the theoretically ideal value is not found.

From the perspective of F7, GOA, SCGOA, and LGOA failed to fail to locate the theoretical ideal value and have not jumped out of the local optimal value. However, as can be observed from the figure, NGOA1 and WOGOA convergence speed is poor, indicating that their performance in the ability to conduct global searches is subpar.

From the perspective of F8-F9, WOGOA's optimization ability is better than the other four algorithms. However, convergence speed is slow on F8 and fast on F9.

In general, after comparing with several different improved GOA, it can still prove the superiority of WOGOA optimization and further highlight the optimization ability of WOGOA.

5. Conclusion

This research suggests a hybrid grasshopper optimization algorithm that incorporates the whale optimization method to address the shortcomings of the sluggish convergence rate and low precision of GOA. Based on the GOA, this algorithm introduces the Levy flight's whale spiral bubble net hunting behavior as well as logistic chaotic mapping and nonlinear parameters and weights to improve the algorithm's capacity to run out of the local optimum as well as its ability to do local searches and accelerate convergence. According to the experimental data, WOGOA can deliver outcomes that are highly competitive in terms of convergence ability and accuracy.

WOGOA has achieved good results in testing functions, so applying WOGOA to practical optimization problems, especially complex, dynamic, and large-scale optimization problems, is another research direction of this article. We can create binary and multi-objective WOGOA versions in subsequent work. And it can also be applied to other fields, such as photovoltaic maximum power tracking, power load prediction, and robot path shortest planning, and it is expected that this algorithm will achieve good results.

The abbreviations involved in this article are shown in Table 4

Table 4

Abbreviate table

Full name	Abbreviated name	Full name	Abbreviated name
Hybrid grasshopper optimization algorithm	WOGOA	Ant lion optimization algorithm	ALO
Improved whale optimization algorithm	IWOA	Harris hawks optimization	HHO
Modified whale optimization algorithm	MWOA	Cauchy mutation grasshopper optimization with trigonometric substitution	SCGOA
Whale optimization algorithm	WOA	Levy flight based grasshopper optimization algorithm	LGOA
Grasshopper optimization algorithm	GOA	The first type of new locust optimization algorithm	NGOA1

Acknowledgments

This research was funded by the Short-term Power Load Forecasting based on Feature Selection and optimized LSTM with DBO which is the fundamental scientific research project of Liaoning Provincial Department of Education (JYTMS20230189), and the Application of hybrid Grey Wolf Algorithm in job shop scheduling problem of the Research Support Plan for Introducing High-Level Talents to Shenyang Ligong University (No. 1010147001131).

REFERENCES

[1]. *H. Deng, L. Liu, J. Fang and Q. Huang*, "A novel improved whale optimization algorithm for optimization problems with multi-strategy and hybrid algorithm", in Math. Comput. Simulat., **vol. 205**, Sept. 2023, pp. 794-817

[2]. *S. Suman, D. Chatteriee, M. Anand, R. Mohanty and A. Bhattacharya*, "An improved Harmonic reduction technique for the PV-WIND hybrid generation scheme using modified whale optimization algorithm (MWOA)", in Ser. C-Electr. Eng. Co. Sci., **vol. 84**, no.1, Sept. 2022, pp. 231-248

[3]. *L. Zaaraoui, A. Mansouri and N. Smairi*, "NMOPSO: An improved multiobjective PSO algorithm for permanent magnet motor design", in Ser. C-Electr. Eng. Co. Sci., **vol. 84**, no.1, Sept. 2022, pp. 201-214

[4]. *S. Mirjalili and A. Lewis*, "The whale optimization algorithm", in Adv. Eng. Softw., **vol. 95**, Sept. 2016, pp. 51-67

[5]. *S. Saremi, S. Mirjalili and A. Lewis*, "Grasshopper optimization algorithm: theory and application", in Adv. Eng. Softw., **vol. 105,** Sept. 2017, pp. 30-47

[6]. *Y. Lan*, "Grasshopper optimization algorithm based on chaos and cauchy mutation and Feature selection", in Microelectron. Comput., **vol. 38**, no. 11, Sept. 2021, pp. 21-30

[7]. *S. C. Phatak, S. S. Rao*, "Logistic map: A possible random-number generator", in Phys. Rev. E, **vol. 51**, no. 4, Sept. 1995, pp. 3670

[8]. *Y. Li, and L. Gu*, "Grasshopper optimization algorithm based on curve adaptation and simulated annealing", in Comput. Appl. Res., **vol. 36**, no. 12, Sept. 2019, pp. 3637-3643

[9]. *Z. Gao, H. Zhang, Y. Chen, J. Liu, and Z. Nie*, "Mutation Motion Tracking Based on Dynamic Weighted Grasshopper Optimization Algorithm", in J. Zhengzhou Univ. (Sci. Edit.), **vol. 52**, no. 2, Sept. 2020, pp. 36-44

[10]. *Z. Lammouchi, and K. Barra*, "Particle swarm weighting factor optimisation for predictive control of three level inverter with balanced voltages", in Int. J. Power Electr., **vol. 12**, no. 1, Sept. 2020, pp. 302-316

[11]. *S. Mirjalili*, "The ant lion optimizer", in Adv. Eng. Softw., **vol. 83**, Sept. 2015, pp. 80-98

[12]. *A. A. Heidari, S. Mirjalili, H. Faris, L. Aljarah, M. Mafarja and H. Chen*, "Harris hawks optimization: Algorithm and applications", in Future Gener. Comp. Sy., **vol. 97**, Sept. 2019, pp. 849-872

[13]. *S. Zhao, P. Wang, A.A. Heidari, X. Zhao, C. Ma and H. Chen*, "An enhanced Cauchy mutation grasshopper optimization with trigonometric substitution: engineering design and feature selection", in Eng Comput. Germany., **vol. 38**, no. 5, Sept. 2022, pp. 4583-4616

[14]. *D. Mokeddem*, "Parameter extraction of solar photovoltaic models using enhanced levy flight based grasshopper optimization algorithm", in J. Electr. Eng. Technol., **vol. 16**, no. 1, Sept. 2021, pp. 171-79

[15]. *Q. Wang and F. Li*, "Two new types of grasshopper optimization algorithms for solving high-dimensional complex functions", in J. Chongqing Univ. Technol. (Nat. Sci.), **vol. 35,** no. 12, Sept. 2021, pp. 277-283