A Nonlinear African Vulture Optimization Algorithm Combining Henon Chaotic Mapping Theory and Reverse Learning Competition Strategy

Authors: Baiyi Wang, Zipeng Zhang, Patrick Siarry, Xinhua Liu, Grzegorz Królczyk, Dezheng Hua, Frantisek Brumercik, Zhixiong Li, Xinhua Liu

Date: 2024-03-28T00:00:00+00:00

Abstract

As a novel intelligent optimization algorithm, the African Vultures Optimization Algorithm (AVOA) has been widely applied across various domains. However, when addressing complex multimodal problems, the AVOA exhibits several limitations, including low search accuracy, deficient search capability, and a tendency to converge to local optima. To mitigate these primary deficiencies, a nonlinear African Vultures Optimization Algorithm integrating Henon chaotic mapping theory and a reverse learning competition strategy (HWEAVOA) is proposed. First, the Henon chaotic mapping theory and an elite population strategy are employed to enhance the randomness and diversity of the initial vulture population. Furthermore, a nonlinear adaptive incremental inertial weight factor is introduced during the location update phase to rationally balance exploration and exploitation capabilities, thereby preventing individuals from becoming trapped in local optima. The reverse learning competition strategy is designed to expand the search space for optimal solutions and enhance the ability to escape local optima. The HWEAVOA and other state-of-the-art comparative algorithms are utilized to solve classical benchmark functions and CEC2022 test functions. Compared with other algorithms, the convergence curves of HWEAVOA descend more rapidly and exhibit smoother trajectories. These experimental results demonstrate that the proposed HWEAVOA ranks first across all test functions, surpassing the comparative algorithms in terms of convergence speed, optimization capability, and solution stability. Meanwhile, HWEAVOA achieves a comparable level of algorithmic complexity, and its overall performance is competitive among swarm intelligence algorithms.

Full Text

A Nonlinear African Vulture Optimization Algorithm Combining Henon Chaotic Mapping Theory and Reverse Learning Competition Strategy

Baiyi Wang[1], **Zipeng Zhang**[1], **Patrick Siarry**[2], **Xinhua Liu**[1]***, Grzegorz Królczyk***[3]***, Dezheng Hua***[1]***, Frantisek Brumercik***[4]***, Zhixiong Li***[3]

[1] China University of Mining and Technology, Xuzhou, China
[2] University Paris-Est Créteil Val de Marne, 61 Av. du General de Gaulle, 94010, Creteil, France
[3] Department of Manufacturing Engineering and Automation Products, Opole University of Technology, 45-758 Opole, Poland
[4] Department of Design and Machine Elements, Faculty of Mechanical Engineering, University of Zilina, Univerzitna 1, 010 26 Zilina, Slovakia

Corresponding authors: liuxinhua@cumt.edu.cn; zhixiong.li@yonsei.ac.kr (Zhixiong Li)

Abstract

As a new intelligent optimization algorithm, the African Vultures Optimization Algorithm (AVOA) has been widely applied in various fields today. However, when solving complex multimodal problems, AVOA still exhibits certain shortcomings, such as low search accuracy, deficient search capability, and a tendency to fall into local optima. To alleviate these main drawbacks, this paper proposes a nonlinear African vulture optimization algorithm combining Henon chaotic mapping theory and reverse learning competition strategy (HWEAVOA). First, the Henon chaotic mapping theory and elite population strategy are introduced to improve the randomness and diversity of the initial vulture population. Furthermore, a nonlinear adaptive incremental inertial weight factor is incorporated into the position update phase to rationally balance exploration and exploitation capabilities while preventing individuals from falling into local optima. The reverse learning competition strategy is designed to expand the discovery field for optimal solutions and strengthen the ability to escape local optima. HWEAVOA and other advanced comparison algorithms are employed to solve classical and CEC2022 test functions. Compared with other algorithms, the convergence curves of HWEAVOA drop faster and exhibit smoother trajectories. These experimental results demonstrate that the proposed HWEAVOA ranks first across all test functions, showing superiority over comparison algorithms in terms of convergence speed, optimization capability, and solution stability. Meanwhile, HWEAVOA achieves a general level in algorithmic complexity, and its overall performance is competitive among swarm intelligence algorithms.

Keywords: African vultures optimization algorithm; Henon chaotic mapping theory; Nonlinear adaptive incremental inertial weight factor; Reverse learning competition strategy

1. Introduction

Optimization problems are ubiquitous in many fields, such as intelligent production, scientific research, and economic management. Today, the complexity and difficulty of optimization issues are increasing, with solutions becoming more dynamic and computationally demanding. Finding one or more points in a multidimensional hyperspace is often necessary. Traditional data processing methods are increasingly inadequate for coping with the data surge problems brought about by the digital age. Therefore, intelligent optimization algorithms are needed to determine accurate solutions for us (Kar, 2016; D. Karaboga & Akay, 2009; Li, Wang, & Gandomi, 2021). Intelligent optimization algorithms represent a new class of optimization methods that simulate biological behaviors and certain natural physical phenomena. With the application of these algorithms, many complex optimization problems can be solved efficiently (Valdez, Castillo, Cortes-antonio, & Melin, 2022). Compared with traditional optimization algorithms, intelligent optimization algorithms converge rapidly, are robust, pervasive, and stable (Cui, Geng, Zhu, & Han, 2017; Nabaei, et al., 2018). In recent years, intelligent optimization algorithms have developed unprecedentedly, with scholars proposing a series of new methods such as supply-demand-based optimization (Zhao, Wang, & Zhang, 2019), carnivorous plant algorithm (Meng, Pauline, & Kiong, 2021), honey badger optimization algorithm (Hashim, Houssein, Hussain, Mabrouk, & Al-Atabany, 2022), Runge Kutta optimization algorithm (Ahmadianfar, Heidari, Gandomi, Chu, & Chen, 2021), hunger game search algorithm (Yang, Chen, Heidari, & Gandomi, 2021), wild horse optimization algorithm (Naruei & Keynia, 2022), material generation optimization algorithm (Oyelade, Ezugwu, Mohamed, & Abualigah, 2022), spider jumping optimization algorithm (Peraza-Vazquez, et al., 2022), reptile search algorithm (Abualigah, Abd Elaziz, Sumari, Geem, & Gandomi, 2022), and capuchin search algorithm (Braik, Sheta, & Al-Hiary, 2021). Intelligent optimization is widely used in system identification (N. Karaboga, 2009), path planning (Wang, Yan, & Gu, 2019), troubleshooting (Deng, Li, Li, Chen, & Zhao, 2022), neural networks (Xu, Yu, & Gulliver, 2021), optimization control (Hamza, Yap, & Choudhury, 2017), and other fields (Kalinli & Karaboga, 2005; N. Karaboga & Cetinkaya, 2011) due to its good searchability.

The African Vultures Optimization Algorithm (AVOA) (Abdollahzadeh, Gharehchopogh, & Mirjalili, 2021) is a new intelligent optimization algorithm proposed by Benyamin Abdollahzadeh et al. in 2021, which simulates the foraging and navigation behaviors of African vultures. The algorithm has the advantages of simple structure, easy implementation, and outstanding performance in finding optimal values, and has been well applied in various fields. Salah et al. (2022) introduced AVOA to optimize the PID controller and applied it to control DC microgrid voltage. Mekala, Sumathi, and Shobana (2022) proposed a multi-polymer charging scheduling strategy based on AVOA to realize proper planning of electric vehicle charging. Singh, Houssein, Mirjalili, Cao, and Selvachandran (2022) introduced the African vulture algorithm to optimize solutions to the traveling salesman shortest path problem. Diab, Tolba, El-Rifaie, and Denis (2022) introduced the African vulture algorithm to accurately predict unknown parameters of various solar photovoltaic units. Zhang, Khayatnezhad, and Ghadimi (2022) applied the African vulture algorithm to an actual PEMFC baseline case study and established an optimal evaluation model for fuel cells.

Although AVOA has numerous applications, it still shows deficiencies in search capability and a tendency to fall into local optima. To address these problems, Liu et al. (2022) introduced quasi-antagonistic learning mechanisms, differential evolution operators, and adaptive parameters to balance AVOA's exploration and exploitation capabilities. Fan, Li, and Wang (2021) used chaotic mapping and time-varying tools to optimize the global optimal solution and convergence performance of AVOA. Soliman, Hasanien, Turky, and Muyeen (2022) proposed a new African vulture-grey wolf hybrid optimizer to improve the algorithm's convergence speed and stability. Kannan, Mannathazhathu, and Raghavan (2022) proposed a hybrid optimization algorithm based on honey badger and African vulture, implementing global optimization search and reducing the probability of falling into local optima.

To date, due to its novelty, there are few studies on improvements to AVOA. Although existing improved AVOA variants increase optimization performance, they still have limitations and uncertainties, such as lack of diversity in initialized populations, imbalance between exploration and exploitation capabilities, and waste of valuable population information, resulting in the AVOA algorithm being sensitive to local optima and unable to obtain ideal solutions. Therefore, an improved AVOA with multi-strategy (HWEAVOA) is proposed to eliminate the uncertainty and restrictions of the original AVOA and subsequently improve its performance. First, Henon chaotic mapping theory and elite population strategy (HCE) are proposed, which make the initial population distribution more homogeneous, enhancing the global optimization performance and convergence rate of AVOA. Then, the nonlinear adaptive incremental inertial weight factor (NWF) is introduced to optimally update the position of vultures. This strategy assists vulture populations in searching at different convergence rates, balancing exploration and exploitation abilities, and effectively prevents AVOA from falling into local optima. Finally, the reverse learning competition strategy (RLC) is designed to increase population diversity. Poorly performing vulture individuals are given learning opportunities and have the probability to become dominant individuals. This strategy expands the discovery field for optimal solutions and avoids the generation of local optimum phenomena.

To evaluate the effectiveness of HWEAVOA, classical and CEC2022 test functions are used to compare the optimization performance of AVOA and other improved algorithms.

The rest of this paper is organized as follows. Section 2 briefly introduces the original AVOA. Section 3 describes the technical details of the HWEAVOA algorithm. Section 4 reliably analyzes the performance of HWEAVOA using classical and CEC2022 test functions. Section 5 summarizes this study by discussing the results and proposing potential areas for future investigation.

---

2. African Vultures Optimization Algorithm (AVOA)

The African Vultures Optimization Algorithm (AVOA) simulates the foraging and navigation behaviors of African vultures based on their natural lifestyle. Each individual in the population relies on its hunger rate for corresponding behaviors and completes the switch between exploration and development stages. The hunger rate is calculated as follows:

$$F = \left(\frac{t}{T}\right) \times \left(\sin\left(\frac{\pi}{2} \times \frac{t}{T}\right) + \cos\left(\frac{\pi}{2} \times \frac{t}{T}\right)\right) \times W \times \text{rand}$$

where F is the vulture hunger rate, t indicates the current number of iterations, T is the maximum number of iterations, W shows a fixed parameter set before the algorithm works, and rand represents a random number between -1 and 1.

When the F value is greater than 1, the vulture searches for food in different regions and enters the exploration phase, using the following formula to search for food in other areas:

$$P_1 = \text{rand}$$

where P_1 is a control parameter with values between 0 and 1, and rand represents a random number between 0 and 1.

Regarding the optimal solution guidance strategy, the remaining vultures search for food near one of the optimal vultures at a random distance. The position update formula is as follows:

$$X_i^{t+1} = R(i) \times \left(X_i^t - F \times P_1 \times (X_i^t - \text{BestV}_p)\right)$$

where X_i^{t+1} represents the vulture position vector in the next iteration, F is the hunger rate of vulture individuals in the current iteration, $R(i)$ is a place where vultures move randomly, which is used as a coefficient vector to increase random motion and obtained using the formula where rand is a random number between 0 and 1; X_i^t indicates the current vector position of the vulture; BestV_p indicates the best vulture chosen at random, and the solution formula is as follows:

$$\text{BestV}_p = \begin{cases} \text{BestV}_1 & \text{if } P_i \geq P_{\text{rand}} \\ \text{BestV}_2 & \text{if } P_i < P_{\text{rand}} \end{cases}$$

where BestV_1 and BestV_2 represent the two best adapted vultures in the vulture population respectively; L_1 and L_2 represent parameters between 0 and 1 which are to be measured respectively, and their sum is 1; P_i represents the probability of selecting the best vulture.

On the other hand, vultures perform random search strategies with the following positional update formula:

$$X_i^{t+1} = R(i) \times (X_i^t - F \times \text{rand}_2 \times (\text{ub} - \text{lb}) \times \text{rand}_3 + \text{lb})$$

where rand_2 and rand_3 are random values between 0 and 1, and ub and lb represent the upper and lower bounds of the variable.

If the value of F is less than 1, the vulture enters the development phase and looks for food near the best solution. When $0.5 \leq F \leq 1$, as shown in Fig. 1 [Figure 1: see original paper], the population obtains food through the implementation of the conflict profit strategy and rotational flight strategy. The two methods are selected and executed by the following formula:

$$P_2 = \text{rand}$$

where P_2 is a control parameter with values between 0 and 1, and rand is a random number between 0 and 1.

Regarding the conflict profitability strategy, weak vultures try to obtain food by causing conflicts between healthy vultures to fatigue them, and their position update formula is as follows:

$$X_i^{t+1} = \text{BestV}_p \times \left(\left|X_i^t - \text{BestV}_p\right| \times F \times \text{rand}_4 + \text{BestV}_p\right)$$

where rand_4 is a random number between 0 and 1.

In addition, the vulture rotational flight strategy is as follows:

$$X_i^{t+1} = \text{BestV}_p \times \left(\left|X_i^t - \text{BestV}_p\right| \times F \times \text{rand}_5 \times \cos\left(\frac{2\pi}{T}\right) + \text{BestV}_p\right)$$

where rand_5 and rand_6 are random numbers between 0 and 1; S_1 and S_2 are calculated by formula (8). Finally, the vulture position update is completed by formula (9). When $F < 0.5$, as shown in Fig. 2 [Figure 2: see original paper], the population obtains food through the implementation of individual competition strategy and population competition strategy, with two methods selected and executed by formula (11).

$$P_3 = \text{rand}$$

where P_3 is a control parameter with values between 0 and 1, and rand is a random number between 0 and 1.

When an individual competition strategy is implemented, multiple vultures may accumulate on the same food source, and the position update formula is as follows:

$$\text{BestV}_p = \begin{cases} \text{BestV}_1 \times (A \times \text{BestV}_1 - \text{BestV}_2) \times F & \text{if } P_i \geq P_{\text{rand}} \\ \text{BestV}_2 \times (A \times \text{BestV}_1 - \text{BestV}_2) \times F & \text{if } P_i < P_{\text{rand}} \end{cases}$$

where $A = \frac{P_1+P_2}{2}$.

When a population competition strategy is implemented, multiple vultures may accumulate on the same food source, and the position update formula is as follows:

$$X_i^{t+1} = \left| X_i^t - \text{BestV}_p \right| \times F \times \text{Levy} + \text{BestV}_p$$

where Levy indicates the levy flight.

---

3. The Proposed Optimization Algorithm (HWEAVOA)

As mentioned above, although the overall mechanism of the AVOA algorithm is simple and easy to implement, it still has limitations such as lack of diversity in initialized populations, imbalance between exploration and exploitation capabilities, and waste of valuable population information, which results in the AVOA algorithm being sensitive to local optima and unable to obtain ideal solutions. This section proposes three improvement strategies: the Henon chaotic mapping theory and elite population strategy (HCE), the nonlinear adaptive incremental inertial weight factor (NWF), and the reverse learning competition strategy (RLC), which are discussed in the following sections.

3.1 Henon Chaotic Mapping Theory and Elite Population Strategy (HCE) The primitive vulture population is initialized by randomization, resulting in uneven distribution and lack of population diversity. Due to low randomness, primitive AVOA always faces high uncertainty. Therefore, if homogeneous initialization of the population can be achieved, population diversity can be effectively increased and the searching efficiency of the algorithm can be improved. To this end, this paper introduces Henon chaotic mapping theory and elite population strategy.

Henon chaotic mapping theory is a nonlinear theory characterized by nonlinearity, initial sensitivity, randomness, and ergodicity. The Henon chaotic map is produced in 2-dimensional space as a typical discrete chaotic map. Its kinetic formula is as follows:

$$\begin{cases} x_{n+1} = 1 - a x_n^2 + y_n \\ y_{n+1} = b x_n \end{cases}$$

where the state of the Henon chaotic map is determined by four parameters a, b, x_0, and y_0, which is more complex than 1-dimensional chaotic maps. This paper takes $a = 1.4$ and $b = 0.3$ to ensure strong randomness of the generated chaotic sequence when the function enters the chaotic state. By the above mapping method, the chaotic mapping initialization population is obtained.

Then, the elite population strategy is adopted, which combines the chaotic mapped initialized population and the conventional initialized population, calculates the fitness of each initial vulture, and sorts them. Finally, the first N elite individuals are selected, and the sequence of privileged individuals is as follows:

$$\{x_1, x_2, \dots, x_i, \dots, x_N\}$$

where N is the number of vultures in the population.

These improved approaches make the initial population distribution more homogeneous and provide the initial population with more possibilities, which enhances the global optimization performance and convergence rate of AVOA.

3.2 Nonlinear Adaptive Incremental Inertial Weight Factor (NWF)
In AVOA, the entire vulture population starts from global search and gradually returns to local search, mechanically performing the exploration and exploitation phases under the leadership of the optimal vulture and suboptimal vulture. However, the whole algorithmic process is not static. It is difficult to effectively balance the global search stage and local search stage of the population using the original position update formula of AVOA. The entire process ignores the actual environment of the vulture population, which causes AVOA to be sluggish in convergence and prone to local optima. Therefore, this paper introduces the nonlinear adaptive incremental inertial weight factor to realize rational allocation between the global exploration phase and local exploitation phase in different evolutionary periods.

The nonlinear adaptive incremental inertial weight factor is added to the process of position renewal of vulture populations, calculated as follows:

$$\omega = \begin{cases} \alpha + \beta \times \sin\left(\frac{\pi}{2} \times \frac{t}{T}\right) \times \text{rand} & \text{if } \operatorname{mod}(t,6) \geq 3 \\ \alpha + \beta \times \text{rand} & \text{if } \operatorname{mod}(t,6) \leq 3 \end{cases}$$

where α and β are selection factors for the initial optimal vulture and secondary vulture, rand is a random number between 0 and 1, and t is the residual number of iterations divided by T. The nonlinear adaptive incremental inertial weight factor is then introduced into the vulture position update formulas in the exploration and development phases, as shown in formulas (18), (19), and (20).

$$X_i^{t+1} = \begin{cases} \omega \times \text{Equation}(14) & \text{if } P_{\text{rand}} \geq P_1 \\ \omega \times \text{Equation}(16) & \text{if } P_{\text{rand}} < P_1 \end{cases}$$

$$X_i^{t+1} = \begin{cases} \omega \times \text{Equation}(8) & \text{if } P_{\text{rand}} \geq P_2 \\ \omega \times \text{Equation}(10) & \text{if } P_{\text{rand}} < P_2 \end{cases}$$

$$X_i^{t+1} = \begin{cases} \omega \times \text{Equation}(12) & \text{if } P_{\text{rand}} \geq P_3 \\ \omega \times \text{Equation}(13) & \text{if } P_{\text{rand}} < P_3 \end{cases}$$

The position of vultures is optimally updated through the above formulas. This step considers evolutionary differences between population vultures during evolution, which adaptively confers inertial weight factors of different magnitudes. When the inertial weight factor increases, the global optimization ability of the algorithm is significantly enhanced; however, the local search ability is reduced and solution accuracy could be lower. When the inertial weight factor declines, the global optimization ability decreases while the local optimization ability is enhanced and solution accuracy is higher. The inertial weight factor assists vultures in searching at different convergence rates until they approach the optimal value in the next iteration. The inertial weight factor improves the search ability of the vulture population at different stages while maintaining unchanged overall search behavior. In the early stage, when the population has strong global search ability, its local search ability is improved appropriately, increasing the search accuracy and convergence rate of AVOA. In the later stage, when the population has strong local search ability, its global search ability is improved appropriately, effectively preventing AVOA from falling into local optima. These characteristics meet the needs of AVOA for global exploration and local exploitation abilities at different evolutionary times.

3.3 Reverse Learning Competition Strategy (RLC) The original AVOA simply uses information from the optimal location in the iteration process, easily wasting part of the valuable information in the population. Therefore, this paper introduces the reverse learning competition strategy. The main idea of the reverse competitive learning strategy is to increase population diversity and the chance of obtaining better solutions by simultaneously exploring both the positive and negative directions of the search space. As shown in Fig. 3 [Figure 3: see original paper], poorly performing vulture individuals are given more learning opportunities, and they have the probability to become dominant individuals. In this way, the loss of useful information is well addressed and the accuracy of the next input is effectively guaranteed.

Based on each output solution, the reverse learning solution is obtained through the reverse learning competition strategy. The calculation formula is as follows:

$$PE_i^{t+1} = \text{rand} \times (\text{ub} + \text{lb}) - P_i^{t+1}$$

where rand is a random number between 0 and 1.

The output position of vultures in this iteration is optimized by calculating the population fitness P_i^{t+1} and PE_i^{t+1}. In this way, the optimal individual position is not lost, and the optimal individual information can be used to significantly improve the robustness of the AVOA algorithm.

In summary, problems in the original AVOA algorithm such as uneven population initialization distribution, lack of population diversity, no reasonable balance between global and local search stages, and easy loss of better personal information are well solved based on three strategies: HCE, NWF, and RLC, corresponding to the Henon chaotic mapping theory and elite population strategy, the nonlinear adaptive incremental inertial weight factor, and the reverse learning competition strategy, respectively. The framework of the HWEAVOA algorithm is shown in Fig. 4 [Figure 4: see original paper] and its technical details are described in Table 1 .

---

4. Experimental Results and Discussion

Test experiments based on classical and CEC2022 test functions are carried out to validate the performance of HWEAVOA in this section. The mathematical expressions and function characteristics of classical and CEC2022 test functions are given in Appendix 1 and Appendix 2 respectively.

All experiments were conducted on Windows 10 (64 bit) running on an Intel Core i7 CPU with 16GB RAM, utilizing MATLAB 2021a. To minimize algorithmic randomness, each algorithm accumulated 1000 tests on the classical test functions, with population size and maximum iterations for all algorithms set to 30 and 500, respectively. In CEC2022, each algorithm accumulated 30 tests, the population size was set to 20, and the maximum iterations for all algorithms were set to 200,000 and 1,000,000 for 10 and 20 dimensions, respectively.

The test functions are evaluated by these algorithms, and the search performance and optimization performance of each algorithm are assessed by comparing the test results with the optimal values of the functions. The results mainly contain average (Avg) and standard deviation (Std), where the average verifies the optimization ability of the algorithm and the standard deviation verifies the stability of the optimization process. The data shown in Table 2 , Table 4 , Table 5 , Table 6 , Table 7 , Table 8 , Table 9 , Table 10 , Table 11 , Table 12 , Table 13 , and Table 14 represent the results of the comparison optimization algorithms in solving these benchmark function problems.

In this section, HWEAVOA is compared with the original AVOA, six AVOA variant algorithms based on three improved strategies, and ten advanced intelligent optimization algorithms to verify its superior optimization performance.

4.1 Parameters Tuning and Analysis In HWEAVOA, W (used in Eq. (1)) is mainly used to balance the exploration and development stages of vulture populations, while P_1 (used in Eq. (18)), P_2 (used in Eq. (19)), and P_3 (used in Eq. (20)) are mainly used to assist vultures in selecting different behaviors and updating their positions. In the original AVOA, these four parameters are derived from AVOA and set as 2.5, 0.6, 0.4, and 0.6, respectively. This section aims to study the effects of different parameters on HWEAVOA performance. The value of W is increased from 2 to 3 with a step size of 0.5, while the values of P_1, P_2, and P_3 are increased from 0.4 to 0.6 with a step size of 0.1. This paper adopts the control variable method, changing only one parameter in each experiment while setting other values according to AVOA defaults. For example, when W increases from 2 to 3, P_1, P_2, and P_3 are set as 0.6, 0.4, and 0.6. When P_1 increases from 0.4 to 0.6, W, P_2, and P_3 are set as 2.5, 0.4, and 0.6. When P_2 increases from 0.4 to 0.6, W, P_1, and P_3 are set as 2.5, 0.6, and 0.6. When P_3 increases from 0.4 to 0.6, W, P_1, and P_2 are set as 2.5, 0.6, and 0.4. The effects of different parameters on HWEAVOA performance are shown in Table 2 .

The results show that HWEAVOA achieves the highest search ability and algorithm stability when W, P_1, P_2, and P_3 are set as 2.5, 0.6, 0.4, and 6, respectively. Therefore, HWEAVOA selects the same parameter settings as AVOA in this paper.

Based on the above analysis, the main parameters of HWEAVOA and its variant algorithms are shown in Table 3 .

4.2 Effects of HCE, NWF, and RLC To study the optimization performance of HWEAVOA, three improved strategies (HCE, NWF, and RLC) are introduced into the optimization process. This section focuses on the impact of these three improvement strategies on the original AVOA. The strategies are abbreviated as “H,” “W,” and “E.” The performance of HWEAVOA and six AVOA variants—HAVOA, WAVOA, EAVOA, HWAVOA, HEAVOA, WEAVOA—and AVOA is tested. HAVOA, WAVOA, and EAVOA represent the introduction of HCE, NWF, and RLC into the original AVOA algorithm, respectively. HWAVOA means introducing both HCE and NWF into AVOA.

HEAVOA means introducing both HCE and RLC into AVOA. WEAVOA means introducing both NWF and RLC into AVOA. The parameter settings of these variant algorithms are shown in Table 3.

As shown in Table 4, experimental results indicate that optimal values for functions F1–F4 can be obtained by HWEAVOA in unimodal benchmark functions. For multimodal benchmark functions, HWEAVOA obtains optimal values for functions F8 and F11, while the optimal values found by HWEAVOA are the best among all algorithms for other multimodal benchmark functions. For fixed-dimension multimodal benchmark functions, HWEAVOA obtains optimal values for functions F16, F17, F19, F21, F22, and F23, while HWEAVOA performs better than different algorithms on other fixed-dimensional multimodal benchmark functions.

As shown in Fig. 5 [Figure 5: see original paper], some convergence curves are displayed for 0–50 generations to show iterative differences more clearly and intuitively. When solving F1 and F4, HWEAVOA's convergence speed is close to WAVOA, HWAVOA, and WEAVOA, significantly faster than HAVOA and EAVOA. In solving F6–F8, F12, F13, and F15, HWEAVOA's convergence speed is slightly faster than other variant algorithms. When solving F9, all algorithms obtain the optimal value around 30 iterations, but HWEAVOA's convergence rate remains the fastest. When solving F21 and F23, the convergence speeds of HAVOA, EAVOA, HEAVOA, WEAVOA, AVOA, and HWEAVOA are significantly faster than WAVOA and HWAVOA. Meanwhile, among HAVOA, EAVOA, HEAVOA, WEAVOA, AVOA, and HWEAVOA, HWEAVOA's convergence speed is slightly faster than HAVOA, EAVOA, HEAVOA, WEAVOA, and AVOA. In summary, HWEAVOA demonstrates advantages in convergence speed and precision when solving classical test functions compared with other variants.

4.3 Evaluation with Classical Test Functions HWEAVOA's performance is compared with ten advanced intelligent optimization algorithms to verify its superiority, including GA (Deng, Zhang, et al., 2022), PSO (Cheng & Jin, 2015), DE (Das & Suganthan, 2011), GWO (Mirjalili, Mirjalili, & Lewis, 2014), COOT (Naruei & Keynia, 2021), RSO (Dhiman, Garg, Nagar, Kumar, & Dehghani, 2021), ATO (Naruei & Keynia, 2021), AOA (Abualigah, Diabat, Mirjalili, Abd Elaziz, & Gandomi, 2021), IHAOAVOA (Xiao, et al., 2022), and OAVOA (Jena, Naik, Panda, & Abraham, 2022). Algorithm parameter settings are shown in Table 3. Comparative results based on 23 functions are shown in Table 5. For fairness, all optimization methods run under the same test conditions.

Experimental results in Table 5 show that HWEAVOA ranks first in 18 test functions: F1–F7, F9–F12, F15–F17, F19, and F21–F23, including seven unimodal benchmark functions, four multimodal benchmark functions, and seven fixed-dimension multimodal benchmark functions.

HWEAVOA's performance in classical test functions indicates its ability in solving real-world optimization problems. Especially compared with IHAOAVOA and OAVOA, HWEAVOA performs better than IHAOAVOA in seven benchmark functions (F5–F8, F12, F15, and F18) and better than OAVOA in F1–F7, F12, F13, and F15, benefiting from the NWF strategy that balances exploration and exploitation capabilities like a powerful engine driving HWEAVOA to excavate small areas effectively.

For more obvious analysis, the Friedman test is used to compare solution results. As shown in Table 6, HWEAVOA's overall average ranking is 1.70, lower than IHAOAVOA's 1.96 and OAVOA's 2.43. Meanwhile, HWEAVOA demonstrates much faster convergence speed than other advanced algorithms, as shown in Fig. 6 [Figure 6: see original paper]. When solving F1, F3, F5, F7, F12, and F13, HWEAVOA converges fastest, followed by IHAOAVOA, significantly faster than other advanced algorithms, meaning it reaches the best area very quickly. This shows that HWEAVOA's search performance, solution precision, and convergence velocity are significantly enhanced. These features mean HWEAVOA has robust global search capability, making an appropriate balance between search mechanisms.

4.4 Evaluation with CEC2022 Test Functions After comprehensively evaluating HWEAVOA using classical test functions, CEC2022 test functions (Dimension = 10 and 20) are applied for further comprehensive evaluation. This section aims to provide a reference for subsequent research, enabling other scholars to compare and evaluate AVOA based on CEC2022 test functions. For fairness, all algorithms perform under the same test conditions. Algorithm parameter settings are shown in Table 3.

In CEC2022 test functions, F1 is the unimodal function, F2–F5 are basic functions, F6–F8 are hybrid functions, and F9–F12 are composition functions. Different function types evaluate algorithm performance from different aspects. HWEAVOA's experimental results with 10 and 20 dimensions are shown in Table 7 and Table 8.

Results in Tables 7 and 8 show that HWEAVOA consistently obtains competitive results in almost all test functions for both 10 and 20 dimensions. For the entire CEC2022 test suite, HWEAVOA's advantages are obvious, as the HCE strategy makes the initial population distribution more homogeneous and enhances AVOA's global optimization performance and convergence rate. Meanwhile, the RLC strategy increases population diversity and the chance of obtaining better solutions, giving poorly performing individuals opportunities to become dominant.

For more comprehensive analysis, the Friedman test compares solution results. As shown in Table 9, HWEAVOA's overall average ranking is 2.25 in 10 dimensions, lower than COOT's 2.83 and OAVOA's 3.00. In 20 dimensions, HWEAVOA's overall average ranking is 2.50, far lower than COOT's 3.75 and GTO's 4.17. In summary, HWEAVOA overcomes AVOA's shortcomings and achieves better algorithm performance with the assistance of HCE and RLC strategies.

4.5 Population Size Analysis This section examines the influence of population size on classical test functions. To adequately analyze HWEAVOA's population size sensitivity, the population size is set to 30, 60, 100, and 200 to demonstrate its influence.

Experimental data in Table 10 show that HWEAVOA's Avg and Std values vary slightly across population sizes in 23 test functions, but the overall order of magnitude remains basically at the same level. Because there is little difference between the given population sizes, the earlier claim is supported. Therefore, HWEAVOA is more stable when population changes.

4.6 Scalability Analysis This section investigates the scalability of HWEAVOA and AVOA using classical test functions with different dimensions: 10, 30, 50, 100, and 1000. Experimental results using thirteen classical functions (F1–F13) with different dimension sizes are shown in Table 11.

When solving functions F1–F4, F9, F10, and F11, HWEAVOA obtains optimal values across all five dimension settings. Although optimal values for other functions in different dimensions cannot be obtained, HWEAVOA's search ability and algorithm stability remain better than AVOA. As shown in Fig. 7 [Figure 7: see original paper], HWEAVOA dramatically improves convergence speed and solving accuracy in the solution space when dealing with problems in different dimensions compared with original AVOA. Meanwhile, as shown in Tables 6 and 7, HWEAVOA consistently ranks in the top three positions when solving CEC2022 test functions F1–F4, F7–F10, and F12. This analysis shows HWEAVOA has better ability to handle high-dimensional problems than original AVOA.

4.7 Time Consumption and Algorithm Complexity Analysis To study HWEAVOA's time consumption, the running time of HWEAVOA and other advanced algorithms in classical test functions is shown in Table 12, with measurement units in seconds.

Experimental results show HWEAVOA's time consumption is similar to original AVOA and inferior to some classical optimization algorithms, but its algorithm accuracy and stability are leading. These results demonstrate that HWEAVOA's overall search performance has not been affected. Therefore, HWEAVOA has higher search efficiency than other advanced algorithms.

Moreover, algorithm complexity is also an important evaluation criterion for optimization algorithms. Therefore, algorithm complexity is calculated based on CEC2022 test functions. According to Sun, Sun, Li, & Ieee (2022), the calculation rules are as follows:

1. Run the test program below:

```
x = 0.55
for i = 1:200000
    x = x + x
    x = x / 2
    x = x * x
    x = sqrt(x)
    x = log(x)
    x = exp(x)
    x = x / (x + 2)
```

2. Evaluate time consumption for CEC2022 test function F1 with 200,000 evaluations of dimension D, denoted as T_1.
3. Evaluate complete time for mentioned algorithms with 200,000 evaluations of the same D-dimensional F1, denoted as T_2.
4. Calculate T_2 for 5 independent runs: $T_2 = \text{mean}(T_2)$.

Finally, algorithm complexity is calculated as $(T_2 - T_1)/T_0$. Using this method, algorithm complexity for HWEAVOA and other improved algorithms is shown in Table 13.

Table 13 indicates that HWEAVOA is outperformed by some advanced algorithms in algorithm complexity when Dimension = 10 and 20. However, HWEAVOA can achieve relatively competitive results with very small iterations throughout the convergence process, leading other algorithms in overall performance. Overall, combined with previous research (Sun, Li, Huang, & Ieee, 2022), HWEAVOA's results have reached the general level of swarm intelligence algorithms and remain competitive in algorithm complexity.

4.8 Wilcoxon Rank Sum Test Although test results from classical and CEC2022 test functions show HWEAVOA's superiority to some extent, due to the stochastic nature of meta-heuristic algorithms, it is important to further illustrate significant differences among these algorithms statistically. The Wilcoxon rank sum test (Manalo, Biermann, Patil, & Mehta, 2022) can count significant differences between algorithms and is often used to evaluate optimization performance of improved algorithms. Therefore, the Wilcoxon rank sum test is applied to CEC2022 test functions in 10 and 20 dimensions to further verify HWEAVOA at significance level $\alpha < 0.05$.

Statistical results of the Wilcoxon rank sum test of HWEAVOA relative to each comparison algorithm are shown in Table 14, where "+", "=", and "-" respectively indicate HWEAVOA is superior, equal, or worse than comparison algorithms. A p-value less than 0.05 indicates HWEAVOA shows a more significant difference than the compared algorithm.

According to test results in Table 14, compared with other advanced algorithms, HWEAVOA's test p-values are all less than 0.05, and most symbols are "+".

Therefore, HWEAVOA is statistically superior to other advanced algorithms, with significant differences between HWEAVOA and these algorithms.

---

5. Conclusions and Future Works

To address issues of original AVOA (i.e., tendency to fall into local optima and imbalance between global and local search stages), this paper proposes an improved African vultures optimization algorithm (HWEAVOA) with three efficient optimization strategies. First, the Henon chaotic mapping theory and elite population strategy improve the randomness and diversity of the initial vulture population. Furthermore, the nonlinear adaptive incremental inertial weight factor introduced in the location update phase satisfies requirements for exploration and exploitation abilities in different phases and prevents the algorithm from falling into local optima. The addition of the reverse learning competition strategy allows the algorithm to expand the discovery field for optimal solutions, accelerate convergence speed, and strengthen the ability to escape local optima. In simulation tests, HWEAVOA and other advanced comparison algorithms solve classical and CEC2022 test functions. Through comparative analysis of experimental results and convergence curves, HWEAVOA's optimization ability and convergence speed for solving complex functions are obviously better than other advanced algorithms. Meanwhile, HWEAVOA reaches a general level in algorithmic complexity, and its overall performance is competitive among swarm intelligence algorithms.

Future possible works are as follows: Although HWEAVOA improves algorithm performance in optimization ability, convergence speed, and solution stability, it still has room for improvement in time consumption and algorithmic complexity. Follow-up work will further optimize HWEAVOA for these issues. Furthermore, large-scale problems and dynamic problems represent current development trends in swarm intelligence, but many algorithms perform poorly and are prone to local optimization when solving these problems. We will continue studying based on HWEAVOA to improve the application space of swarm intelligence algorithms.

CRediT Authorship Contribution Statement

Baiyi Wang: Conceptualization, Methodology, Analysis, Data Curation, Writing - review & editing, Funding acquisition.
Zipeng Zhang: Methodology, Supervision, Data collection, Writing – original draft.
Patrick Siarry: Investigation, Data collection, Writing – review & editing.
Xinhua Liu: Supervision, Funding acquisition.
Grzegorz Królczyk: Analysis, Writing – review & editing.
Dezheng Hua: Analysis, Writing - Review & Editing.

Frantisek Brumercik: Data collection, Visualization.
Zhixiong Li: Investigation, Project administration, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

All data that produce the results in this work can be requested from the corresponding author.

Acknowledgements

The support of the National Natural Science Foundation of China (No. 51975568), the Independent Innovation Project of "Double-First Class" Construction of China University of Mining and Technology (2022ZZCX06), and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX23_{2681}) in carrying out this research are gratefully acknowledged. The research leading to these results has received funding from the Norwegian Financial Mechanism 2014-2021 under Project Contract No. 2020/37/K/ST8/02748.

References

Abdollahzadeh, B., Gharehchopogh, F. S., & Mirjalili, S. (2021). African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Computers & Industrial Engineering*, 158, 37.

Abualigah, L., Abd Elaziz, M., Sumari, P., Geem, Z. W., & Gandomi, A. H. (2022). Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Systems with Applications*, 191, 33.

Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The Arithmetic Optimization Algorithm. *Computer Methods in Applied Mechanics and Engineering*, 376, 38.

Ahmadianfar, I., Heidari, A. A., Gandomi, A. H., Chu, X. F., & Chen, H. L. (2021). RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. *Expert Systems with Applications*, 181, 22.

Braik, M., Sheta, A., & Al-Hiary, H. (2021). A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm. *Neural Computing & Applications*, 33, 2515-2547.

Cheng, R., & Jin, Y. C. (2015). A social learning particle swarm optimization algorithm for scalable optimization. *Information Sciences*, 291, 43-60.

Cui, Y. F., Geng, Z. Q., Zhu, Q. X., & Han, Y. M. (2017). Review: Multi-objective optimization methods and application in energy saving. *Energy*, 125, 681-704.

Das, S., & Suganthan, P. N. (2011). Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transactions on Evolutionary Computation*, 15, 4-31.

Deng, W., Li, Z. X., Li, X. Y., Chen, H. Y., & Zhao, H. M. (2022). Compound Fault Diagnosis Using Optimized MCKD and Sparse Representation for Rolling Bearings. *IEEE Transactions on Instrumentation and Measurement*, 71, 9.

Deng, W., Zhang, X. X., Zhou, Y. Q., Liu, Y., Zhou, X. B., Chen, H. L., & Zhao, H. M. (2022). An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems. *Information Sciences*, 585, 1-20.

Dhiman, G., Garg, M., Nagar, A., Kumar, V., & Dehghani, M. (2021). A novel algorithm for global optimization: Rat Swarm Optimizer. *Journal of Ambient Intelligence and Humanized Computing*, 12, 8457-8482.

Diab, A. A. Z., Tolba, M. A., El-Rifaie, A. M., & Denis, K. A. (2022). Photovoltaic parameter estimation using honey badger algorithm and African vulture optimization algorithm. *Energy Reports*, 8, 384-393.

Fan, J. H., Li, Y., & Wang, T. (2021). An improved African vultures optimization algorithm based on tent chaotic mapping and time-varying mechanism. *PLOS ONE*, 16, 52.

Hamza, M. F., Yap, H. J., & Choudhury, I. A. (2017). Recent advances on the use of meta-heuristic optimization algorithms to optimize the type-2 fuzzy logic systems in intelligent control. *Neural Computing & Applications*, 28, 979-999.

Hashim, F. A., Houssein, E. H., Hussain, K., Mabrouk, M. S., & Al-Atabany, W. (2022). Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems. *Mathematics and Computers in Simulation*, 192, 84-110.

Jena, B., Naik, M. K., Panda, R., & Abraham, A. (2022). A novel minimum generalized cross entropy-based multilevel segmentation technique for the brain MRI/dermoscopic images. *Computers in Biology and Medicine*, 151, 36.

Kalinli, A., & Karaboga, N. (2005). A new method for adaptive IIR filter design based on tabu search algorithm. *AEU-International Journal of Electronics and Communications*, 59, 111-117.

Kannan, S. N., Mannathazhathu, S. E., & Raghavan, R. (2022). A novel compression based community detection approach using hybrid honey badger African vulture optimization for online social networks. *Concurrency and Computation-Practice & Experience*, 34, 17.

Kar, A. K. (2016). Bio inspired computing - A review of algorithms and scope of applications. *Expert Systems with Applications*, 59, 20-32.

Karaboga, D., & Akay, B. (2009). A survey: algorithms simulating bee swarm intelligence. *Artificial Intelligence Review*, 31, 61-85.

Karaboga, N. (2009). A new design method based on artificial bee colony algorithm for digital IIR filters. *Journal of the Franklin Institute-Engineering and Applied Mathematics*, 346, 328-348.

Karaboga, N., & Cetinkaya, M. B. (2011). A novel and efficient algorithm for adaptive filtering: Artificial bee colony algorithm. *Turkish Journal of Electrical Engineering and Computer Sciences*, 19, 175-190.

Li, W., Wang, G. G., & Gandomi, A. H. (2021). A Survey of Learning-Based Intelligent Optimization Algorithms. *Archives of Computational Methods in Engineering*, 28, 3781-3799.

Liu, R. J., Wang, T. L., Zhou, J., Hao, X. X., Xu, Y., & Qiu, J. Z. (2022). Improved African Vulture Optimization Algorithm Based on Quasi-Oppositional Differential Evolution Operator. *IEEE Access*, 10, 95197-95218.

Manalo, T. A., Biermann, H. D., Patil, D. H., & Mehta, A. (2022). The Temporal Association of Depression and Anxiety in Young Men With Erectile Dysfunction. *Journal of Sexual Medicine*, 19, 201-206.

Mekala, K., Sumathi, S., & Shobana, S. (2022). A Multi-Aggregator Based Charge Scheduling in Internet of Electric Vehicles Using Fractional African Vulture Sail Fish Optimization. *Cybernetics and Systems*, 27, 1-15.

Meng, O. K., Pauline, O., & Kiong, S. C. (2021). A carnivorous plant algorithm for solving global optimization problems. *Applied Soft Computing*, 98, 40.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, 69, 46-61.

Nabaei, A., Hamian, M., Parsaei, M. R., Safdari, R., Samad-Soltani, T., Zarrabi, H., & Ghassemi, A. (2018). Topologies and performance of intelligent algorithms: a comprehensive review. *Artificial Intelligence Review*, 49, 79-103.

Naruei, I., & Keynia, F. (2021). A new optimization method based on COOT bird natural life model. *Expert Systems with Applications*, 183, 25.

Naruei, I., & Keynia, F. (2022). Wild horse optimizer: a new meta-heuristic algorithm for solving engineering optimization problems. *Engineering with Computers*, 38, 3025-3056.

Oyelade, O. N., Ezugwu, A. E. S., Mohamed, T. I. A., & Abualigah, L. (2022). Ebola Optimization Search Algorithm: A New Nature-Inspired Metaheuristic Optimization Algorithm. *IEEE Access*, 10, 16150-16177.

Peraza-Vazquez, H., Pena-Delgado, A., Ranjan, P., Barde, C., Choubey, A., & Morales-Cepeda, A. B. (2022). A Bio-Inspired Method for Mathematical

Optimization Inspired by Arachnida Salticidade. *Mathematics*, 10, 32.

Salah, B., Hasanien, H. M., Ghali, F. M. A., Alsayed, Y. M., Aleem, S., & El-Shahat, A. (2022). African Vulture Optimization-Based Optimal Control Strategy for Voltage Control of Islanded DC Microgrids. *Sustainability*, 14, 26.

Singh, N., Houssein, E. H., Mirjalili, S., Cao, Y. K., & Selvachandran, G. (2022). An efficient improved African vultures optimization algorithm with dimension learning hunting for traveling salesman and large-scale optimization applications. *International Journal of Intelligent Systems*, 37, 12367-12421.

Soliman, M. A., Hasanien, H. M., Turky, R. A., & Muyeen, S. M. (2022). Hybrid African vultures-grey wolf optimizer approach for electrical parameters extraction of solar panel models. *Energy Reports*, 8, 14888-14900.

Sun, B., Li, W., Huang, Y., & Ieee. (2022). Performance of Composite PPSO on Single Objective Bound Constrained Numerical Optimization Problems of CEC 2022. In *IEEE Congress on Evolutionary Computation (CEC)*. Padua, Italy: IEEE.

Sun, B., Sun, Y. F., Li, W., & Ieee. (2022). Multiple Topology SHADE with Tolerance-based Composite Framework for CEC2022 Single Objective Bound Constrained Numerical Optimization. In *IEEE Congress on Evolutionary Computation (CEC)*. Padua, Italy: IEEE.

Valdez, F., Castillo, O., Cortes-antonio, P., & Melin, P. (2022). APPLICATIONS OF INTELLIGENT OPTIMIZATION ALGORITHMS AND FUZZY LOGIC SYSTEMS IN AEROSPACE: A REVIEW. *Applied and Computational Mathematics*, 21, 233-245.

Wang, X. W., Yan, Y. X., & Gu, X. S. (2019). Spot welding robot path planning using intelligent algorithm. *Journal of Manufacturing Processes*, 42, 1-10.

Xiao, Y. N., Guo, Y. L., Cui, H., Wang, Y. W., Li, J., & Zhang, Y. P. (2022). IHAOAVOA: An improved hybrid aquila optimizer and African vultures optimization algorithm for global optimization problems. *Mathematical Biosciences and Engineering*, 19, 10963-11017.

Xu, L. W., Yu, X., & Gulliver, T. A. (2021). Intelligent Outage Probability Prediction for Mobile IoT Networks Based on an IGWO-Elman Neural Network. *IEEE Transactions on Vehicular Technology*, 70, 1365-1375.

Yang, Y. T., Chen, H. L., Heidari, A. A., & Gandomi, A. H. (2021). Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Systems with Applications*, 177, 34.

Zhang, J. L., Khayatnezhad, M., & Ghadimi, N. (2022). Optimal model evaluation of the proton-exchange membrane fuel cells based on deep learning and modified African Vulture Optimization Algorithm. *Energy Sources Part A-Recovery Utilization and Environmental Effects*, 44, 287-305.

Zhao, W. G., Wang, L. Y., & Zhang, Z. X. (2019). Supply-Demand-Based Optimization: A Novel Economics-Inspired Algorithm for Global Optimization. *IEEE Access*, 7, 73182-73206.

Appendix 1

Classical Test Functions

Function Type	Function Equation	Range	Optimal
Unimodal Benchmark Functions			
F1	$\sum_{i=1}^{n} x_i^2$	[-100,100]	0
F2	$\sum_{i=1}^{n} \|x_i\| + \prod_{i=1}^{n} \|x_i\|$	[-10,10]	0
F3	$\sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	[-100,100]	0
F4	$\max_i\{\|x_i\|, 1 \leq i \leq n\}$	[-100,100]	0
F5	$\sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30,30]	0
F6	$\sum_{i=1}^{n}(x_i + 0.5)^2$	[-100,100]	0
F7	$\sum_{i=1}^{n} ix_i^4 + \mathrm{random}[0,1)$	[-1.28,1.28]	0
Multimodal Benchmark Functions			
F8	$\sum_{i=1}^{n} -x_i \sin(\sqrt{\|x_i\|})$	[-500,500]	-418.9829 × n
F9	$\sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12,5.12]	0
F10	$-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	[-32,32]	0
F11	$\frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600,600]	0
F12	$\frac{\pi}{n}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	[-50,50]	0
F13	$0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	[-50,50]	0

Function Type	Function Equation	Range	Optimal
Fixed-Dimension Multimodal Benchmark Functions			
F14	$\left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	[-65,65]	1
F15	$\sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right]^2$	[-5,5]	0.00030
F16	$4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	[-5,5]	-1.0316
F17	$(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1-\frac{1}{8\pi})\cos(x_1) + 10$	[-5,5]	0.398
F18	$[1+(x_1+x_2+1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_{1x}2 + 3x_2^2)] \times [30+(2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_{1x}2 + 27x_2^2)]$	[-2,2]	3
F19	$-\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	[0,1]	-3.86
F20	$-\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	[0,1]	-3.32
F21	$\sum_{i=1}^{4} \left[(X-a_i)(X-a_i)^T + c_i\right]^{-1}$	[0,10]	-10
F22	$\sum_{i=1}^{4} \left[(X-a_i)(X-a_i)^T + c_i\right]^{-1}$	[0,10]	-10
F23	$\sum_{i=1}^{4} \left[(X-a_i)(X-a_i)^T + c_i\right]^{-1}$	[0,10]	-10

Appendix 2

CEC2022 Test Functions

Function Type	Functions
Unimodal Function	Shifted and full Rotated Zakharov Function
Basic Functions	Shifted and full Rotated Rosenbrock's FunctionShifted and full Rotated Expanded Schaffer's f6 FunctionShifted and full Rotated Non-Continuous Rastrigin's FunctionShifted and full Rotated Levy Function

Function Type	Functions
Hybrid Functions	Hybrid Function 1 (N = 3)Hybrid Function 2 (N = 6)Hybrid Function 3 (N = 5)
Composition Functions	Composition Function 1 (N = 5)Composition Function 2 (N = 4)Composition Function 3 (N = 5)Composition Function 3 (N = 6)

Search range: [-100, 100]^D

Note: Figure translations are in progress. See original paper for figures.