International Journal for Housing Science and Its Applications
Publish August 6, 2025. Volume 46, Issue 3 Pages 7023-7036

https://doi.org/10.70517/ijhsa463602

A Study on Strategy Enhancement of Intelligent Algorithm-Based Automated Design Tools for Complex Design Tasks

Yu'an Ning[1] and Kiesu Kim[1,*]
[1] Department of Industrial Design, Silla University, Busan, 46958, South Korea
Corresponding authors: (e-mail: kks@silla.ac.kr).

Abstract Complex design tasks require the support of efficient intelligent optimization algorithms, and the traditional beluga optimization algorithm is defective in convergence speed and optimization stability. In this study, we propose a multi-strategy hybrid improved beluga whale algorithm (MHIBWO) that integrates the MTent population initialization strategy, the step-size adjustment strategy, and the longitudinal and transversal crossover strategy to address the limitations of the traditional beluga optimization algorithms for complex design tasks. The MTent mapping enhances the diversity of the initial populations through the introduction of random numbers; the step-size adjustment strategy optimizes the weight allocation by using the sine-cosine model to improve the global optimization ability; the vertical and horizontal crossover strategy maintains the population diversity through horizontal crossover and vertical crossover operations to prevent premature convergence. Numerical experiments show that the MHIBWO algorithm has significant advantages in single-peak and multi-peak function tests, and the average optimization accuracy and standard deviation of the three test functions under 80-dimensional conditions are 0. Compared with the standard BWO and the other five swarm intelligence optimization algorithms, the MHIBWO algorithm has a faster convergence speed and higher solving accuracy. Among the 18 multitask optimization benchmark tasks, the MHIBWO algorithm has significantly better solution quality than the existing algorithms in 14 tasks, and achieves an accuracy of 2.21 × 10^{-2} on task T2 of the CI+HS problem, which is close to the global optimum. The experimental results demonstrate that the MHIBWO algorithm has excellent global search capability, local development capability and stability in complex design tasks.

Index Terms Moby Dick optimization algorithm, complex design task, MTent population initialization, step size adjustment strategy, vertical and horizontal crossover strategy, multi-task optimization

I. Introduction

Automated design tools based on intelligent algorithms are methods that can realize automated design based on intelligent algorithms and modular architecture [1], [2]. In the design field, the traditional design method mainly relies on manual and simple design tools to realize the design task, but with the expansion of the business scope, the design task is also gradually complex, the traditional design method is no longer applicable, and with the development of artificial intelligence, the application of intelligent algorithms has prompted the design field to develop in the direction of intelligence [3]-[6].

Compared with traditional design methods, automated design tools have several important implications in complex design tasks: (i) Improve design efficiency, automated design can complete multiple design tasks at one time by writing programs or scripts, which reduces the required manual operations and time and greatly improves the design efficiency [7]-[9]. (ii) Reducing design cost, automated design can reduce errors and repetitive work caused by manual operations, reducing the error rate and cost of the design process [10]-[12]. (iii) Improve design quality, automated design can avoid design errors caused by human negligence or fatigue, ensure the accuracy and consistency of the design, and improve the design quality [13], [14]. (iv) Accelerating the speed to market, automated design can quickly generate design solutions and product models, and perform simulation and analysis, which accelerates product development and speed to market [15]-[17].

Optimization algorithms play a crucial role in engineering design, resource scheduling, path planning and other fields. In recent years, swarm intelligence optimization algorithms inspired by biological behaviors in nature have become an important tool for solving complex optimization problems due to their simplicity and ease of implementation, adaptability and robustness. Existing swarm intelligence optimization algorithms include particle swarm algorithm, gray wolf algorithm and whale optimization algorithm, etc. Each of these algorithms has its own characteristics, but when dealing with high-dimensional complex problems, they still face the problems of slow convergence, easy to fall into the local optimal solution, and insufficient solution accuracy. The beluga optimization algorithm is a novel meta-heuristic algorithm that performs global search and local exploitation by simulating the swimming, feeding and whale falling behaviors of beluga whales, which provides better optimization performance while maintaining a simple computational structure. However, the standard beluga algorithm still has deficiencies in initial population diversity, search strategy balance and local development capability, especially when facing high-dimensional complex design tasks, its optimization ability and convergence performance are difficult to meet the practical needs. In order to improve the applicability and effectiveness of the algorithm, researchers have proposed a variety of improvement strategies, such as adaptive adjustment of parameters, initialization of chaotic mapping, and cross-variable operation, etc. These improvements enhance the performance of the algorithm to a certain extent, but most of the improvements are only aimed at a single aspect of the algorithm, and there is a lack of comprehensive consideration of the synergistic optimization of the algorithm in all phases. In complex design task scenarios, optimization algorithms need to have both strong global exploration capability and fine local development capability, as well as the ability to balance knowledge migration between different tasks.

In this study, to address the shortcomings of the standard beluga algorithm, a multi-strategy hybrid improved beluga algorithm incorporating the MTent population initialization strategy, the step size adjustment strategy, and the vertical and horizontal crossover strategy is proposed, starting from the three key aspects of initialization, position update, and population evolution. The MTent population initialization strategy enhances the traversal of the chaotic mapping by introducing a random factor to improve the diversity of the initial population; the step size adjustment strategy adopts a nonlinear decreasing search factor to optimize the weight distribution and enhance the global optimization seeking ability; and the longitudinal and transversal crossing strategy maintains the diversity of the population and prevents the premature convergence through the knowledge sharing among the individuals of the population. Through the synergy of the three strategies, the balance of the algorithm in the exploration and development phases is realized, and the strategic performance of the algorithm in complex design tasks is enhanced. Multiple sets of numerical experiments and application tests are carried out based on the improved algorithm to verify the effectiveness and superiority of the algorithm.

II. Design of Moby Dick Optimization Algorithm Incorporating Multiple Improvement Strategies

In order to realize the strategy enhancement of intelligent automated design tools in complex design tasks, this chapter improves the traditional beluga optimization algorithm by incorporating various improvement strategies such as MTent population initialization strategy, step size adjustment strategy, and vertical and horizontal crossover strategy.

II. A. Moby Dick Algorithm (BWO)

Beluga algorithm [18] is a new meta-heuristic algorithm inspired by the swimming, feeding and whale falling behavior of beluga whales. Meta-heuristic optimization algorithm is as an improvement of heuristic algorithms, which is based on the mechanism of computational intelligence to find the optimal or satisfactory solution of complex optimization problems, and the algorithm possesses a number of advantages, such as flexibility, high efficiency, stability and so on.

II. A. 1) Content of the BWO algorithm

The BWO algorithm mainly consists of the following phases:

(1) Initialization phase: in the initial phase of the algorithm, a set of solutions are randomly generated as the initial positions of the beluga population. Each solution represents a possible solution of the problem and its position corresponds to the coordinates of the solution in the search space.

(2) Exploration phase: in the exploration phase, the beluga group searches by simulating the swimming behavior of beluga whales. Specifically, each individual beluga whale adjusts its swimming speed and direction according to the distance between the current position and the target position in order to gradually approach the target position. This phase is mainly used to find possible optimal solution regions in the search space.

(3) Exploitation phase: in the exploitation phase, the beluga group conducts local search by simulating the feeding behavior of beluga whales. Specifically, each individual beluga whale conducts a detailed search in the neighborhood of its current position to find a more optimal solution. This phase is mainly used to find a more optimal solution within the currently known optimal solution region.

(4) Whale fall phase: the whale fall phase is one of the features of the BWO algorithm, which simulates the whale fall phenomenon that exists in the biological world. In this phase, the algorithm randomly selects a beluga individual and lets other beluga individuals search around it. This helps the algorithm to jump out of the local optimal solution and improve the global search capability.

(5) Iterative update phase: in each iterative step, the algorithm will be updated according to the position and fitness value of the current beluga group. Specifically, individual beluga whales with better fitness values will be given greater weights, thus influencing the search direction of the whole group. This process is carried out iteratively until termination conditions such as reaching the maximum iteration to find the fullness required solution are met.

II. A. 2) BWO algorithm mechanism

The exploration phase ensures global search capability in the design space by randomly selecting beluga whales, and the exploitation phase controls local search in the design space. To simulate the behavior, beluga whales are considered as search agents that move in the search space by changing the position vector. In addition, the probability of a whale fall, which changes the position of the beluga whale, is considered in the BWO.

Due to the population-based mechanism of BWO, beluga whales are considered as search agents and each beluga whale is a candidate solution which is updated during the optimization process. The matrix to the position of the search agent is modeled as:

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,d} \end{bmatrix} \tag{1}$$

where n is the population size of beluga whales and d is the dimension of the design variable. For all belugas, the corresponding fitness values are stored as follows:

$$F_X = \begin{bmatrix} f(x_{1,1}x_{1,2}\cdots,x_{1,d}) \\ f(x_{2,1}x_{2,2}\cdots,x_{2,d}) \\ \vdots \\ f(x_{n,1}x_{n,2}\cdots,x_{n,d}) \end{bmatrix} \tag{2}$$

The BWO algorithm can move from exploration to development depending on the balance factor B_f, which is mathematically modeled as:

$$B_f = B_0\left(1-\frac{t}{2T}\right) \tag{3}$$

where t is the current iteration, T is the maximum number of iterations, and B_0 varies randomly between $(0,1)$ at each iteration. The exploration phase occurs at the equilibrium factor $B_f > 0.5$, while the exploitation phase occurs at $B_f \le 0.5$. As iteration T increases, the range of fluctuation of B_f decreases from $(0,1)$ to $(0,0.5)$, indicating that the probabilities of the exploitation and exploration phases change significantly, while the probability of the exploitation phase increases with increasing iteration T.

The exploration phase of BWO is built by considering the swimming behavior of beluga whales. Beluga whales can perform social behaviors in different poses, e.g., two pairs of beluga whales swimming closely together in a synchronized or mirrored manner. Therefore, the position of the search agent is determined by the swimming of a pair of beluga whales, and the position of the beluga whales is updated as follows:

$$\begin{cases} X_{i,j}^{T+1} = X_{i,p_j}^{t} + (X_{r,p_1}^{t} - X_{i,p_j}^{t})(1+r_1)\sin(2\pi r_2), j = even \\ X_{i,j}^{T+1} = X_{i,p_j}^{t} + (X_{r,p_1}^{t} - X_{i,p_j}^{t})(1+r_1)\cos(2\pi r_2), j = odd \end{cases} \tag{4}$$

where t is the current number of iterations, $X_{i,j}^{T+1}$ is the new position of the i th beluga in the j th dimension, $P_j(j=1,2,\cdots,d)$ is randomly chosen from the d th dimension, and $X_{i,j}^{t}$ is the position of the i th beluga in the j th dimension. X_{i,p_j}^{t} and X_{r,p_1}^{t} are the current positions of the i th and r th randomly selected beluga. r_1 and r_2 are random numbers in the range $(0,1)$ used to augment the stochastic operator in the exploration phase. $\sin(2\pi r_2)$ and $\cos(2\pi r_2)$ are used to average the random numbers between fins. Depending on the dimensions chosen for odd and even numbers, the updated positions reflect the synchronized or mirrored behavior of beluga whales while swimming or diving.

The development phase of BWO was inspired by the feeding behavior of beluga whales. Belugas can forage and move cooperatively based on the location of neighboring belugas. Thus, beluga whales prey on each other by sharing information about each other's positions, considering the best candidate solution and other solutions. The strategy of Levy flight was introduced in the development phase of BWO to improve convergence. It is assumed that they can catch prey using the Levy flight strategy, which is represented by the mathematical model:

$$X_i^{t+1} = r_3 X_{best}^t - r_4 X_i^t + C_1 * L_F * (X_r^t - X_i^t) \tag{5}$$

where t is denoted as the current number of iterations, X_r^t and X_i^t are the current positions of the i th beluga whale and a random beluga whale, X_i^{t+1} is the new position of the i th beluga whale, and X_{best}^t is the best position of the beluga whale. r_3 and r_4 denote random numbers in the range $(0,1)$, and $C_1 = 2r_4(1 - t / T_{\max})$ is used to measure the strength of random jumps in the intensity of Levy flight. L_F is the Levy flight function, which is calculated as follows:

$$L_F = 0.05 * \frac{u * \sigma}{|v|^{1/\beta}} \tag{6}$$

$$\sigma = \left(\frac{\Gamma(1+\beta) * \sin(\pi\beta / 2)}{\Gamma(1+\beta) * \beta * 2^{(\beta-1)/2}} \right)^{1/\beta} \tag{7}$$

where u and v are normally distributed random numbers and β is the default constant 1.5.

The whale-fall phase simulates small changes in the population, and to ensure that the population size remains constant, we use the position of the beluga whale and the step size of the whale's descent to establish the updated position. The mathematical model is represented as:

$$X_i^{t+1} = r_5 X_i^t - r_6 X_r^t + r_7 X_{step} \tag{8}$$

where r_5, r_6 and r_7 are random numbers between $(0,1)$ and X_{step} is the step size of the whale's fall, determined as:

$$X_{step} = (u_b - l_b) \exp\left(-C_2 \frac{t}{T} \right) \tag{9}$$

where C_2 is the step size factor $(C_2 = 2W_f * n)$ associated with the probability of whale descent and population size, and u_b and l_b are the upper and lower bounds of the variables, respectively. It can be seen that the step size is affected by the bounds on the design variables, the number of iterations and the maximum number of iterations.

In this model, the probability of a whale fall (W_f) is computed as a linear function:

$$W_f = 0.1 - 0.005t / T \tag{10}$$

The probability of a whale fall decreased from 0.1 in the initial iteration to 0.05 in the last iteration, suggesting that the danger to beluga whales decreases as they move closer to the food source during the optimization process.

II. B.Improving the Moby Dick Algorithm

The computational complexity is high due to the mutual constraints and influence between the inner and outer objectives in the two-layer optimization model of demand solution. The beluga algorithm still has some defects in convergence speed and stability of optimization search. Therefore, in order to enhance the position diversity of the initial population, avoid local optimality and further improve the performance of the algorithm, this paper proposes a multi-strategy hybrid improvement method based on the MTent population initialization strategy, the step size adjustment strategy and the vertical and horizontal crossover strategy, and then obtains the improved Moby Dick algorithm MHIBWO.

II. B. 1) MTent population initialization strategy

Chaotic mapping has the characteristics of randomness and ergodicity, and the use of chaotic mapping to generate the initial population can effectively improve the BWO initialization population ergodicity and randomness. The commonly used chaotic mappings are Tent mapping and Logistic mapping.

By comparing the results of the two methods, it can be seen that the traversability and randomness of the population generated by Logistic mapping are poor. In contrast, the initial population generated by Tent mapping has stronger randomness. The Tent expression is as follows:

$$X_{n+1}=\begin{cases}2X_n, & X_n\in[0,0.5)\\ 2(1-X_n), & X_n\in[0.5,1)\end{cases} \tag{11}$$

X_n denotes the initial value, $X_n\in(0,1)$, and X_{n+1} is the value after one Tent mapping. To further enhance the traversal of the population, a MTent is proposed by introducing $rand$ random numbers in order to facilitate jumping into the periodic point and entering the chaotic state. The expression is shown in Eq. (12):

$$X_{n+1}=\begin{cases}2(X_n+rand()/20), & X_n\in[0,0.5)\\ 2(1-X_n+rand()/20), & X_n\in[0.5,1]\end{cases} \tag{12}$$

$rand()\in[0,1]$. comparing the two methods reveals that the initial population derived from MTent exhibits better traversal and randomness, resulting in a more diverse population.

II. B. 2) Step size adjustment strategy

In order to avoid the problem of beluga whales falling into local extremes during predation and causing loss of diversity, a step-size adjustment strategy is adopted in the position update to utilize the oscillatory variation property of the positive cosine model to maintain the diversity of individual discoverers, and thus to improve the global searching ability of the BWO. Improvements are made for the basic step-size search factor $r_i=rand(1-t/Iter_{\max})$, and the new nonlinear decreasing search factor is shown in Eq. (13):

$$r_i=A\times\left[1-\left(\frac{t}{Iter_{\max}}\right)^{\eta}\right]^{1/\eta}, i=1,2,\cdots,7 \tag{13}$$

η is the moderating factor, A is the random number $rand$, and $rand\in(0,1)$.

The weight allocation is optimized by improving the step factor. The improved search method shifts the weights' center of gravity forward, reduces the decreasing speed in the early stage, and thus enhances the global optimality searching ability. And the local search ability is enhanced in the late iteration to shorten the time to obtain the optimal solution and improve the convergence speed.

II. B. 3) Cross-cutting strategies

To balance the local search ability and global exploitation ability for BWO and prevent premature convergence. The vertical and horizontal crossover strategy is introduced in the exploration stage to correct the individual and global optimal solutions of the population and to improve the solution accuracy under the premise of ensuring the convergence speed.

(1) Horizontal crossover strategy

The lateral crossover operation is a crossover operation of two different beluga individuals in the population in the same dimension, which makes the different individuals learn from each other and thus improves the global optimization seeking ability. Assuming that lateral crossover is performed on the parent individuals x_i and x_j, the child individuals Mx_i^T and Mx_j^T are generated as follows:

$$\begin{cases}Mx_{i,d}^T=m_1x_{i,d}^T+(1-m_1)x_{j,d}^T+n_1\times(x_{i,d}^T-x_{j,d}^T)\\ Mx_{j,d}^T=m_2x_{j,d}^T+(1-m_2)x_{i,d}^T+n_2\times(x_{j,d}^T-x_{i,d}^T)\end{cases} \tag{14}$$

m_1 and m_2 are random numbers within $[0,1]$ and N_1 and N_2 are random numbers within $[-1,1]$. $x_{i,d}^T$ and $x_{j,d}^T$ are the d th dimension of the parent individuals x_i and x_j, respectively, and $Mx_{i,d}^T$ and $Mx_{j,d}^T$ are the children generated by the crossover of x_i and x_j in the d th dimension, respectively. Individuals of the generated offspring compete with their parents to retain the better adapted individuals. As a result, the number of children in the outer edge space will gradually and linearly decrease, promoting the algorithm to continuously converge to the optimal solution, so as to improve the convergence efficiency and balance the exploration and development ability of the algorithm.

(2) Vertical crossover strategy

Different from the horizontal crossover operation, the vertical crossover operation performs crossover operations on two different dimensions of the global optimal solution x_i^T. Only one of the dimensions is updated at a time, which can help to jump out of the local optimum while not destroying the information contained in the other normal dimension as much as possible. The process of crossing the d_1 and d_2 dimensions of x_i^T is as follows:

$$Mx_{i,d}^T = m \times x_{i,d_1}^T + (1-m)x_{i,d_2}^T \tag{15}$$

The $Mx_{i,d}^T$ are the offspring individuals generated by longitudinal crossover from the d_1 and d_2 dimensions of individuals x_{i,d_1}^T and x_{i,d_2}^T, $m \in [0,1]$. Similarly, the generated offspring individuals compete with their parents to retain the better adapted individuals.

II. C.Improvement of Moby Dick Algorithm Flow

The MHIBWO algorithm optimization process is shown in Figure 1, and its specific steps are as follows:

Step 1: Initialization parameters: initialize the algorithm parameters, set the population size n, the maximum number of iterations $T_{\max}$. Randomize the distribution of the initial positions of individual beluga whales to generate the corresponding fitness values.

Step 2: Population initialization: MTent mapping is used to initialize the beluga population and randomly generate the individual beluga population, and update the position of beluga whales according to equation (12).

Step 3: Fitness calculation: initial fitness values of beluga populations and individuals are calculated and compared to find the best individual.

Step 4: Update on exploration and exploitation phases: decide whether the beluga whale enters the exploration or exploitation phase based on the equilibrium factor B_f. Make the search factor in Eq. (13) with an improved step size.

(1) If $B_f > 0.5$, the update mechanism enters the exploration phase, which is then improved using the vertical and horizontal crossover strategy and the position of the beluga whale is updated by equations (14)~(15).

(2) If $B_f < 0.5$, then update the exploitation phase.

Calculate and order the fitness values of the new position to find the best result in the current iteration.

Step 5: Update of whale fall phase: calculate the probability W_f of whale fall in each iteration and update the mechanism to enter the whale fall phase.

Step 6: Determination: set the termination check condition. If $T > T_{\max}$, stop the loop. Otherwise, jump back to step 4.

III. Algorithm application experiments and analysis of results

To test the effectiveness of the proposed MHIBWO algorithm in complex design tasks, numerical experiments and performance evaluation experiments are conducted in this chapter.

III. A. Numerical experiments with the MHIBWO algorithm

III. A. 1) Experimental design and parameterization

In order to verify the excellence of the MHIBWO algorithm's solving ability, three sets of experiments are designed in this section:

(1) Experiment 1 is the effectiveness analysis of three different strategies improvement of MHIBWO algorithm.

(2) Experiment 2 compares the MHIBWO algorithm with five other swarm intelligence optimization algorithms particle PSO [19], GWO, SSA [20], WOA [21] and two BWO improvement algorithms, A1BWO and A2BWO, for the comparative analysis of the solution accuracy in different dimensions.

(3) Experiment 3 shows the iterative trend of the above 8 algorithms in low dimensions.

The BWO optimization algorithm using the MTent population initialization strategy is abbreviated as MBWO, the BWO optimization algorithm using the step size adjustment strategy is abbreviated as IBWO, and the BWO optimization algorithm using the longitudinal and transversal crossover strategy is abbreviated as HBWO.

The six test functions used are shown in Table 1, where $f_1 \sim f_2$ is a single-peak test function, which is usually used to test the algorithm's local mining ability. $f_3 \sim f_4$ is a multi-peak test function, which is used to test the algorithm's ability to balance the global search and to jump out of local optima. $f_5 \sim f_6$ is a fixed-dimensional multi-peak test function used to compare the performance of different algorithms.

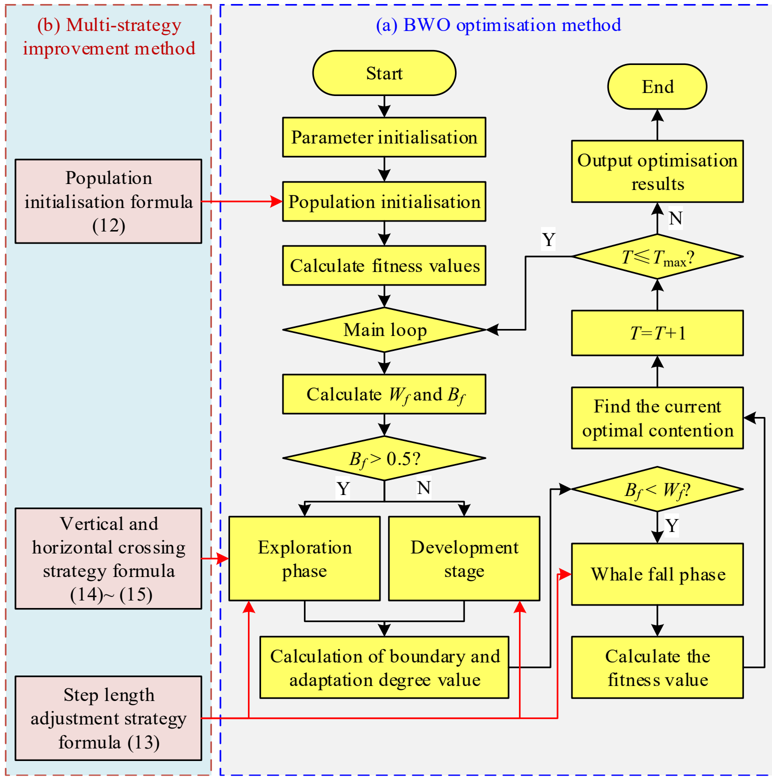


Figure 1: MHIBWO algorithm flowchart

Table 1: Benchmark test functions

Function	Function name	Dimension	Range	Optimal value
f_1	Sphere	50/80/200/800	[-150,150]	0
f_2	Schwefel2.22	50/80/200/800	[-15,15]	0
f_3	Schwefel2.26	50/80/200/800	[-800,800]	-12478.5
f_4	Rastrigin	50/80/200/800	[-6.24,6.24]	0
f_5	Shekel	3	[-64.428,64.428]	1
f_6	Kowalik	5	[-8,8]	0.004

III. A. 2) Analysis of the effectiveness of improvement strategies

In order to test the effectiveness of different improvement strategies, the MHIBWO and IBWO, QBWO algorithms and PBWO algorithms are compared and analyzed with the original BWO algorithm under the conditions of single-peak multiple-peak test function dimensions and fixed-dimension test function, respectively, and the algorithms are analyzed in terms of their superiority with the mean, standard deviation and optimal value as the measurement indexes, and the five algorithms are tested using the benchmark test function in order to prove the effectiveness of each improvement strategy, and the solution results of different improvement strategies are shown in Table 2.

For the single-peak function $f_1 \sim f_2$, the IBWO and HBWO algorithms have obvious advantages, and their accuracies are several tens of orders of magnitude ahead of the initial algorithm, in which the effect of the IBWO algorithm is more advantageous compared to the HBWO algorithm. The experimental results show that these two improved strategies enhance the local mining ability of the original algorithm.

For the multi-peak function $f_3 \sim f_4$, MBWO's shortcomings in the single-peak test function are transformed into advantages here, effectively enhancing the algorithm's solution accuracy, and the MBWO algorithm is still able to show a superior solution ability even in the function which is prone to fall into the local optimum test. The test results show that the nonlinear convergence factor increases with the number of iterations, which effectively improves the ability of the algorithm to jump out of the local area while guaranteeing the search range; at the same time, by adjusting the crossover strategy, the vertical and horizontal crossover strategy is adopted to avoid the algorithm falling into the local optimum at the later stage, which effectively strengthens the algorithm's global search ability.

For the fixed-dimensional multi-peak function $f_5 \sim f_6$, MBWO, IBWO, HBWO algorithms have good robustness, but their advantages are not significant, and their solving accuracy is similar to that of BWO, but the fusion of the improved MIBWO has a significant superiority, and the optimal solutions are all close or equal to the theoretical optimal value.

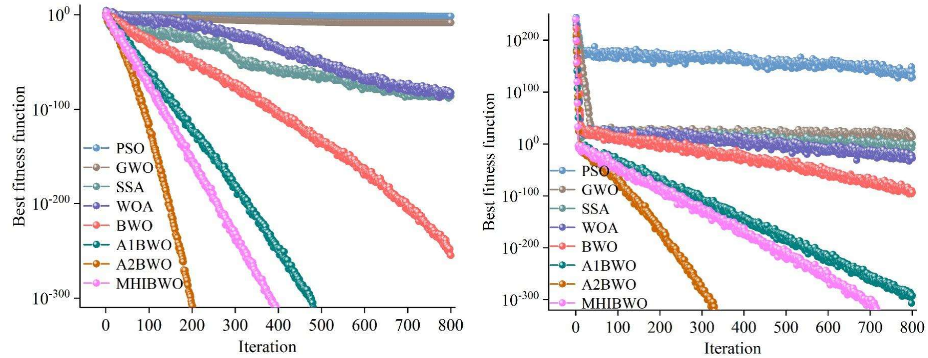
In summary, the MHIBWO algorithm with the integration of the three improvement strategies can well realize the strong global search ability in the early stage of the search, while in the later stage, it has the better ability to jump out of the local optimum, and at the same time, it realizes iterative balance, which strengthens the comprehensive algorithm's ability of searching for the optimal value.

Table 2: Comparison results of different improvement strategies

Function	Algorithm	Mean value	Standard deviation	Optimal value
f_1	BWO	-2.26E-15	3.16E-14	-6.95E-15
	MBWO	-8.27E-29	3.05E-28	-2.44E-29
	IBWO	9.61E-20	6.09E-19	-1.35E-19
	HBWO	2.76E-52	0.00E+00	2.76E-52
	MHIBWO	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_2	BWO	-2.35E-18	3.42E-17	-5.49E-17
	MBWO	-1.27E-33	3.78E-33	-1.16E-34
	IBWO	1.66E-21	3.78E-33	-1.49E-19
	HBWO	4.23E-27	0.00E+00	4.23E-27
	MHIBWO	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_3	BWO	4.37E+02	7.54E+00	4.37E+02
	MBWO	4.26E+02	7.44E+00	4.41E+02
	IBWO	4.35E+02	2.25E+00	4.32E+02
	HBWO	-1.38E+04	0.00E+00	-1.38E+04
	MHIBWO	**-1.25E+04**	**0.00E+00**	**-1.25E+04**
f_4	BWO	2.64E-10	1.38E-9	6.45E-10
	MBWO	-3.41E-10	2.44E-9	-1.83E-10
	IBWO	-8.92E-10	2.56E-9	-6.15E-10
	HBWO	**0.00E+00**	0.00E+00	**0.00E+00**
	MHIBWO	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_5	BWO	-3.26E+01	2.88E-01	-3.30E+01
	MBWO	-1.72E+01	2.38E+01	-3.34E+01
	IBWO	-2.47E+01	1.14E+01	-3.25E+01
	HBWO	1.36E+01	0.00E+00	1.36E+01
	MHIBWO	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_6	BWO	2.11E-01	1.54E-02	2.17E-01
	MBWO	3.35E-01	2.43E-01	3.24E-01
	IBWO	1.85E-01	2.84E-02	1.50E-01
	HBWO	6.55E-04	0.00E+00	6.55E-04
	MHIBWO	**0.00E+00**	**0.00E+00**	**0.00E+00**

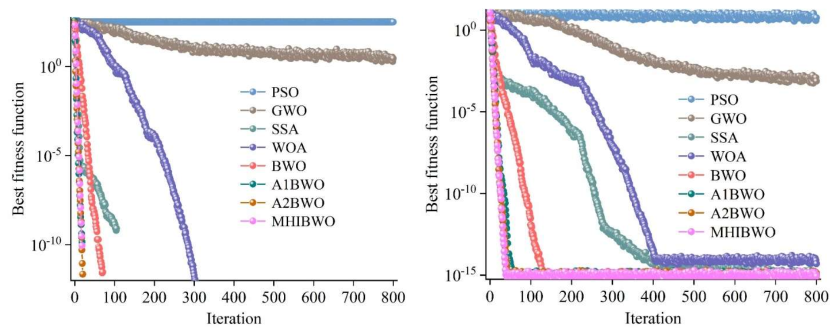
III. A. 3) Comparative analysis of convergence curves

The convergence curves can visualize the optimization process of each algorithm, and also observe the global convergence speed of the algorithms and their ability to jump out of the local extreme points. Setting up the comparison experiment in the dimension D = 80, the maximum number of iterations T = 800, the convergence curve images of the single-peak test function, the multi-peak test function and the multi-peak fixed-dimension test function are plotted as shown in Figs. 2~4, respectively. Among them, Figs. 2(a), 2(b), 3(a), 3(b), 4(a), 4(b) represent the convergence curves of function $f_1 \sim f_6$, respectively.



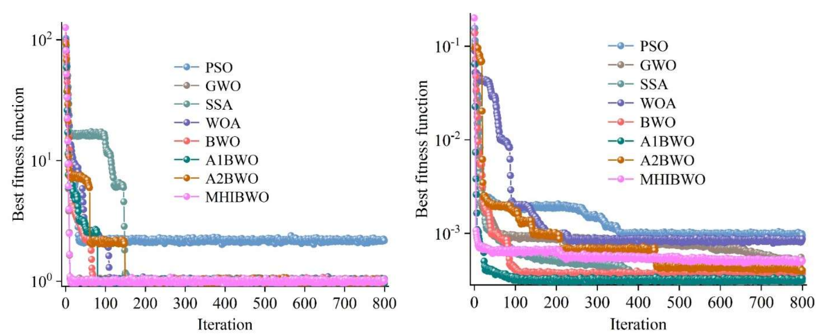
(a) f_1 (b) f_2

Figure 2: Comparison of convergence curves of unimodal test functions



(a) f_3 (b) f_4

Figure 3 Comparison of convergence curves of multimodal test functions



(a) f_5 (b) f_6

Figure 4: Comparison of convergence curves of fixed-dimensional multimodal test functions

The summary analysis of the six convergence curves shows that the optimal algorithm, suboptimal algorithm and the third excellent algorithm appear only between A1BWO, A2BWO and MHIBWO, and at this time, the final scoring system is applied to score the three functions, and the optimal algorithm frequency and its final score can be obtained as shown in Table 3. Among them, K_1, K_2 and K_3 represent the optimal frequency, the second best frequency and the third best frequency, respectively, and the final comprehensive score can be expressed as formula (16):

$$K = 0.6K_1 + 0.3K_2 + 0.1K_3 \tag{16}$$

In this case, the higher score proves that the algorithm performs better.

Observing the data, it can be seen that although the MHIBWO algorithm does not show convergence advantages in all the six test functions, its final score is higher than that of A1BWO and A2BWO, so it can be concluded that the MHIBWO algorithm maintains a certain degree of robustness under the premise of having more excellent convergence ability. For the three types of test functions, the MHIBWO algorithm shows excellent convergence ability with good iteration speed and convergence accuracy.

Table 3: Optimal algorithm frequency

Algorithm	K_1	K_2	K_3	K
A1BWO	3	15	6	6.9
A2BWO	2	6	17	4.7
MHIBWO	18	4	0	12

Table 4: Comparison results of unimodal and multimodal functions in different dimensions

Function	Algorithm	D=80			D=200		
		Mean	SD	Optimal value	Mean	SD	Optimal value
f_1	PSO	-1.67E-01	8.32E-01	-9.75E-01	6.46E-02	1.51E+00	-1.44E+00
	GWO	1.70E-12	2.13E-11	-2.04E-11	-3.63E-10	6.02E-08	5.95E-08
	SSA	-3.84E-37	2.58E-37	-5.53E-37	-1.24E-44	7.19E-44	-3.66E-44
	WOA	-5.97E-39	1.82E-38	8.54E-39	1.15E-39	3.73E-39	1.75E-41
	BWO	1.42E-132	1.81E-131	1.92E-131	-4.51E-134	6.30E-133	6.64E-135
	A1GWO	7.35E-163	0.00E+00	7.74E-163	2.47E-163	0.00E+00	9.49E-165
	A2GWO	-1.16E-164	0.00E+00	-3.12E-164	7.06E-165	0.00E+00	-6.77E-164
	MHIBWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_2	PSO	3.41E-02	5.96E-01	-1.82E-01	1.61E-01	3.04E+00	4.45E+00
	GWO	-8.05E-15	2.92E-14	-2.34E-14	-6.74E-11	7.50E-10	5.36E-10
	SSA	-4.52E-35	2.34E-34	6.83E-35	-9.81E-32	4.84E-29	-1.18E-28
	WOA	-1.18E-54	4.93E-54	-2.41E-59	-2.45E-55	2.48E-54	2.27E-58
	BWO	1.64E-132	2.82E-131	5.41E-132	-3.36E-132	1.63E-131	-1.15E-131
	A1GWO	-1.64E-221	0.00E+00	-1.87E-221	-9.72E-301	0.00E+00	-1.15E-302
	A2GWO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	MHIBWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_3	PSO	-3.37E+01	2.64E+02	-2.93E+01	-6.04E+01	2.74E+02	-3.11E+01
	GWO	2.62E+01	2.40E+02	-3.59E+01	1.53E+01	2.34E+02	4.33E+02
	SSA	1.50E+01	3.75E+02	-3.10E+02	-7.74E+01	3.93E+02	-5.14E+02
	WOA	-3.08E+02	2.52E+01	-3.13E+02	4.14E+02	6.86E+02	4.35E+02
	BWO	4.35E+02	7.76E-06	4.35E+02	4.35E+02	3.92E-05	4.35E+02
	A1GWO	5.28E+02	8.54E-06	5.31E+02	5.36E+02	4.52E-05	5.31E+02
	A2GWO	4.35E+02	5.56E-06	4.35E+02	4.35E+02	7.42E-06	4.35E+02
	MHIBWO	**-2.16E+04**	**0.00E+00**	**-2.16E+04**	**-4.07E+00**	**0.00E+00**	**-4.07E+04**
f_4	PSO	1.74E-01	2.01E+00	1.13E+00	3.34E-02	2.86E+00	1.25E+00
	GWO	2.13E-02	2.53E-01	5.41E-03	2.57E-08	2.96E-07	3.35E-07
	SSA	2.21E-09	8.96E-10	2.04E-09	-7.15E-10	7.83E-13	-7.15E-10
	WOA	6.61E-10	1.26E-09	2.41E-10	-1.66E-09	3.72E-09	-5.94E-12
	BWO	9.88E-10	2.33E-09	-7.92E-10	-1.25E-11	1.24E-09	1.56E-09
	A1GWO	-3.23E-10	1.14E-09	-6.95E-10	-5.53E-09	1.22E-10	1.44E-11
	A2GWO	-1.17E-10	2.72E-09	5.86E-10	8.04E-09	6.12E-10	10.16E-11
	MHIBWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**

III. A. 4) Comparative Analysis of Optimization Accuracy

In order to verify the superiority of the MHIBWO algorithm in terms of optimization seeking ability, six benchmark test functions are selected for experimentation, and the test dimensions are set to 80 and 200, respectively. 8 algorithms with different dimensions of the single-peak function and multi-peak function comparison results are shown in Table 4, and the fixed-dimension function comparison results are shown in Table 5.

For both single-peaked functions $f_1 \sim f_2$ and multi-peaked functions $f_3 \sim f_4$, MHIBWO has the best optimization ability and solution accuracy, and its solution performance is overall better than that of the comparative swarm intelligent optimization algorithms, the standard BWO algorithm, and the literature improved BWO algorithm.

In the 80-dimensional test, the average optimization accuracy, optimal value and standard deviation of MHIBWO on f_1, f_2, f_4 three test functions are 0, which proves that the MHIBWO algorithm has better solving performance in the process of this kind of function, and improves the speed of the solution under the premise of guaranteeing the accuracy of the solution. The f_3 function is also close to the theoretical optimal value, which proves that the MHIBWO algorithm has certain applicability. In the 200-dimensional test, the MHIBWO accuracy increases with the increase of dimension, and the solution results are all 0 except for f_3, which proves that the elevated dimension has not weakened MHIBWO's ability of global exploration and local risk avoidance.

Comparing the test results of fixed-dimensional multi-peak functions of different algorithms, it can be seen that the average and optimal values of MHIBWO algorithm are approximately equal to the theoretical optimal value of the test function, and its standard deviation reaches 0. Compared with the MHIBWO algorithm, the AIBWO algorithm and A2BWO algorithm are not able to converge to the theoretical optimal value, and they do not have a significant advantage over the other swarm intelligent optimization algorithms, thus this proves that the MHIBWO algorithm has good optimization ability and excellent stability for fixed-dimensional multi-peak functions.

Table 5: Comparison results of fixed-dimension functions

Function	Algorithm	Mean	SD	Optimal value
f_5	PSO	-2.53E+01	1.24E+01	-3.34E+02
	GWO	-1.71E+01	2.35E+02	2.34E-02
	SSA	-3.34E+01	3.34E-07	-3.34E+01
	WOA	-1.71E+01	2.35E+01	4.23E-02
	BWO	-3.34E+01	4.05E-02	-3.34E+01
	A1GWO	-3.34E+01	7.48E-07	-3.34E+01
	A2GWO	-3.34E+01	2.34E-03	-3.34E+01
	MHIBWO	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_6	PSO	6.92E+00	8.82E+01	3.20E+00
	GWO	2.37E-01	4.24E+00	-3.98E-01
	SSA	1.72E-01	3.75E-02	1.48E-01
	WOA	1.23E-01	6.99E-02	6.72E-02
	BWO	2.91E-01	1.28E-01	2.63E-01
	A1GWO	1.72E-01	3.75E-02	1.48E-01
	A2GWO	1.68E-01	3.92E-02	1.15E-01
	MHIBWO	**0.00E+00**	**0.00E+00**	**0.00E+00**

III. B. Performance Evaluation Experiments in Complex Design Tasks

This section analyzes the quality of the solutions obtained using the MHIBWO algorithm and evaluates the performance of the algorithm in complex design tasks.

III. B. 1) Quality of solutions

The average objective values and standard deviations obtained from 30 independent runs of the MHIBWO algorithm, MFEA, GA and DE algorithms on all Multi-task Optimization (MTO) benchmarks in this paper are shown in Table 6. In the table, the better results are highlighted in bold, and the experimental results are tested using the Wilcoxon rank sum test with a confidence level of 95%, and are denoted by “+” and “-” to indicate, respectively, that the MHIBWO algorithm of this paper is significantly better and significantly worse than the MFEA.

On all benchmark tasks of MTO, the average objective values of the MHIBWO algorithm using the fusion of multiple improvement strategies are significantly better than the average objective values of GA and DE for a single task, which proves that the MHIBWO algorithm of this paper can improve the optimization performance through better experimental results. Compared with MFEA, the solution quality of this paper's MHIBWO algorithm is significantly better than that of MFEA in 14 out of 18 tasks in terms of average objective value. In task T2 on the CI+HS problem and two tasks on PI+LS, the solution of this paper's MHIBWO algorithm is close to the global optimum 0, while the MFEA only converges to the local optimum. The MFEA and the single-task GA use the same search mechanism and cross-task knowledge migration through probabilistic genetic crossover, so the experimental comparison results further indirectly verify the effectiveness of the MHIBWO algorithm proposed in this paper.

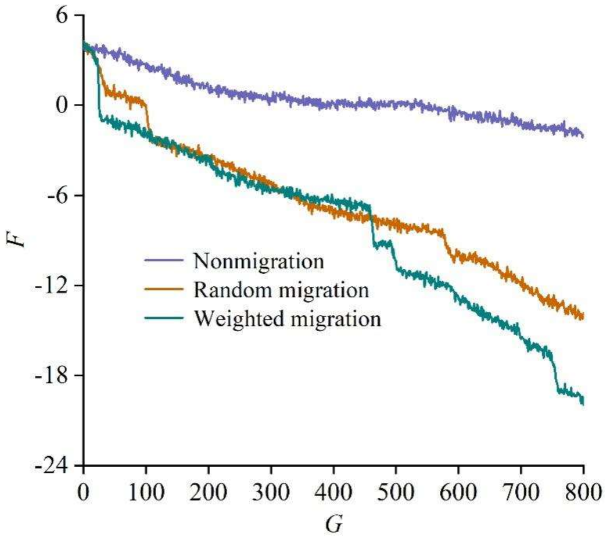
Table 6: Average target values and standard deviations

Problem	Task	Average target value and standard deviation			
		MHIBWO	MFEA	GA	DE
CI+HS	T_1	**1.32×10^{-5}**$\pm4.64\times10^{-5}$ +	$3.73\times10^{-1}\pm6.85\times10^{-2}$	$1.75\times10^{-1}\pm4.49\times10^{-2}$	-
	T_2	**2.21×10^{-2}**$\pm8.46\times10^{-2}$ +	$1.94\times10^{2}\pm4.75\times10^{1}$	-	$1.97\times10^{2}\pm5.51\times10^{1}$
CI+MS	T_1	**2.85×10^{-1}**$\pm5.07\times10^{-1}$ +	$4.55\times10^{0}\pm6.85\times10^{-1}$	$3.24\times10^{0}\pm2.81\times10^{-1}$	-
	T_2	**1.49×10^{1}**$\pm2.38\times10^{1}$ +	$2.23\times10^{2}\pm5.96\times10^{1}$	-	$1.85\times10^{2}\pm5.24\times10^{1}$
CI+LS	T_1	$2.21\times10^{1}\pm4.07\times10^{-1}$ -	**2.13×10^{1}**$\pm5.34\times10^{-2}$	$2.23\times10^{1}\pm4.04\times10^{-2}$	-
	T_2	**3.76×10^{3}**$\pm6.09\times10^{2}$ -	$6.82\times10^{3}\pm9.08\times10^{2}$	-	$6.94\times10^{3}\pm8.41\times10^{2}$
PI+HS	T_1	**3.24×10^{1}**$\pm5.45\times10^{0}$ +	$5.61\times10^{2}\pm9.32\times10^{1}$	$4.42\times10^{2}\pm6.44\times10^{1}$	-
	T_2	$1.85\times10^{2}\pm1.93\times10^{2}$ -	**8.44×10^{0}**$\pm1.89\times10^{0}$	-	$3.15\times10^{2}\pm2.25\times10^{2}$
PI+MS	T_1	**1.69×10^{0}**$\pm4.78\times10^{-1}$ +	$3.73\times10^{0}\pm9.04\times10^{-1}$	$4.25\times10^{0}\pm3.07\times10^{0}$	-
	T_2	**1.49×10^{2}**$\pm5.32\times10^{1}$ +	$6.91\times10^{2}\pm3.52\times10^{2}$	-	$3.45\times10^{5}\pm3.64\times10^{5}$
PI+LS	T_1	**2.27×10^{-5}**$\pm4.73\times10^{-5}$ +	$2.12\times10^{1}\pm9.40\times10^{-2}$	$3.63\times10^{0}\pm5.92\times10^{-1}$	-
	T_2	**2.29×10^{-3}**$\pm3.81\times10^{-3}$ +	$2.18\times10^{1}\pm3.04\times10^{0}$	-	$2.38\times10^{0}\pm1.16\times10^{0}$
NI+HS	T_1	**6.11×10^{1}**$\pm2.88\times10^{1}$ +	$9.94\times10^{2}\pm6.93\times10^{2}$	$1.81\times10^{3}\pm1.44\times10^{3}$	-
	T_2	**2.92×10^{1}**$\pm2.17\times10^{1}$ +	$2.64\times10^{2}\pm6.75\times10^{1}$	-	$2.02\times10^{2}\pm4.81\times10^{1}$
NI+MS	T_1	$8.14\times10^{-1}\pm7.08\times10^{-2}$ +	**4.28×10^{-1}**$\pm6.09\times10^{-2}$	$8.73\times10^{3}\pm8.21\times10^{3}$	-
	T_2	**2.71×10^{0}**$\pm5.09\times10^{-1}$ +	$2.75\times10^{1}\pm3.54\times10^{0}$	-	$1.36\times10^{1}\pm1.95\times10^{0}$
NI+LS	T_1	**3.36×10^{1}**$\pm5.75\times10^{0}$ +	$6.25\times10^{2}\pm1.26\times10^{2}$	$4.26\times10^{2}\pm9.04\times10^{1}$	-
	T_2	$7.14\times10^{3}\pm9.78\times10^{2}$ +	**3.74×10^{3}**$\pm4.64\times10^{2}$	-	$6.67\times10^{3}\pm7.58\times10^{2}$

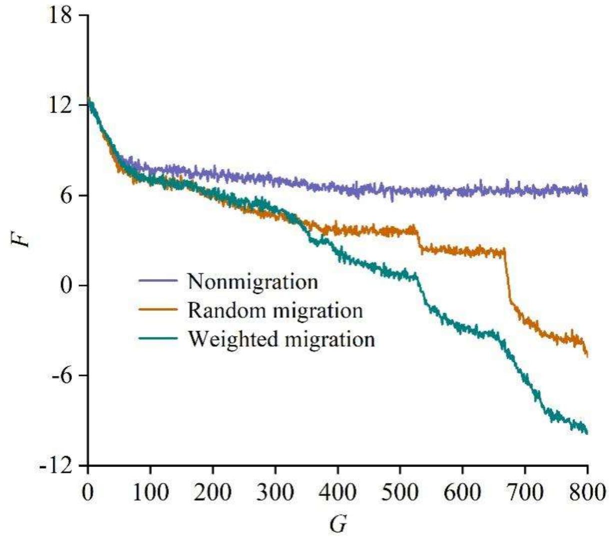
III. B. 2) Performance in complex design tasks

Complex design tasks are usually combined by multiple tasks, in order to further verify the superiority of the performance of the proposed MHIBWO algorithm, this section evaluates the performance of the MHIBWO algorithm in multi-tasking. The six functions in the MTO benchmark are used as the six tasks from task 1 to 6 respectively, and they are optimized simultaneously using this paper's MHIBWO algorithm with GA, DE and PSO as single-task optimizers. At this time, the multi-task optimization problem is more complex compared to the two-task optimization problem, where the cross-task solution migration interval parameter $g=6$ is reset to increase the cross-task knowledge-sharing frequency, and a comparison of the weighted task selection, random task selection, and single-task optimization of this paper's MHIBWO algorithm is shown in Fig. 5. Where, (a)~(c) denote the single-task optimizer based on GA, DE and PSO, respectively.

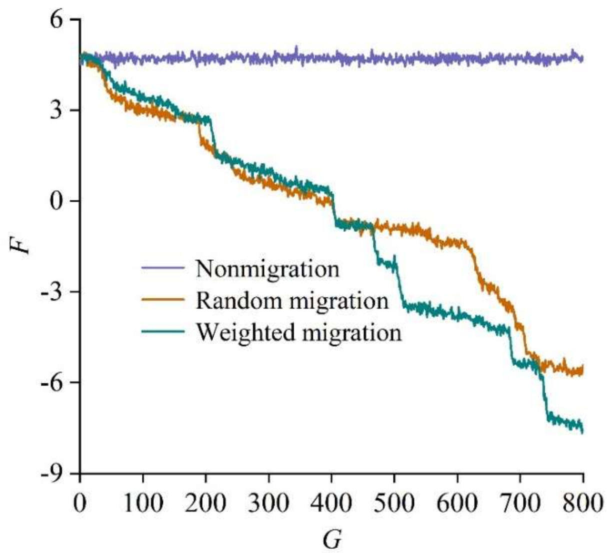
It can be seen that the multitasking optimization algorithm using knowledge migration has significantly better optimization performance for all tasks than the three single-tasking optimization algorithms, while the single-tasking optimization algorithm and the multitasking optimization algorithm use the same basic search mechanism, the reason why the multitasking optimization algorithm achieves a better optimization performance lies in the useful information migrated between its tasks and optimization algorithms, and in the early stages of the optimization process, the paper of weighted task selection In the early stage of the optimization process, the difference in optimization performance between the MHIBWO algorithm with weighted task selection and the MHIBWO algorithm with stochastic task selection is not obvious, but as the number of generations increases, the MHIBWO algorithm with weighted task selection is able to select the appropriate tasks for knowledge sharing, so it has a better performance in terms of searching speed and finding high quality solutions.



(a) GA (T_1)



(b) DE (T_2)



(c) PSO (T_3)

Figure 5: The comparison results of the MHIBWO algorithm

IV. Conclusion

The multi-strategy hybrid improved beluga whale algorithm (MHIBWO) effectively enhances the algorithm's optimization-seeking performance in complex design tasks by incorporating the MTent population initialization strategy, the step-size adjustment strategy, and the longitudinal and transverse crossover strategies. The benchmark function test results show that the MHIBWO algorithm has excellent performance in solving single-peak function, multi-peak function and fixed-dimension multi-peak function, and the average optimization accuracy of the three test functions (f1, f2, f4) in the 80-dimension test reaches 0, with a standard deviation of 0, which is significantly better than that of the comparison algorithm. Algorithm score analysis shows that the final comprehensive score of MHIBWO algorithm reaches 12, which is higher than that of A1BWO (6.9) and A2BWO (4.7), indicating that it has more superior convergence ability and robustness. In the multi-task optimization experiments, the MHIBWO algorithm significantly outperforms the Multifactor Evolutionary Algorithm (MFEA) in terms of solution quality in 14 out of the 18 benchmark tasks, and is close to the globally optimal solution in both tasks of the PI+LS problem, with accuracies of 2.27×10^{-5} and 2.29×10^{-3}. The knowledge migration analysis proves that the MHIBWO algorithm for weighted task selection can efficiently share the knowledge between suitable tasks and improve the overall optimality finding ability. The comprehensive experimental results show that the MHIBWO algorithm achieves a balance between the powerful global search capability in the pre-search stage and the fine local development capability in the post-search stage, providing an efficient and reliable intelligent optimization tool for solving complex design tasks.

References

[1] Wenjie, Y. I., Tingting, L. I. U., & Ben, N. I. U. (2025). Review of Automated Design of Intelligent Optimization Algorithm. INFORMATION AND CONTROL, 54(2), 206-225.

[2] Stützle, T., & López-Ibáñez, M. (2018). Automated design of metaheuristic algorithms. In Handbook of metaheuristics (pp. 541-579). Cham: Springer International Publishing.

[3] Kieran, C., Doorman, M., & Ohtani, M. (2015). Frameworks and principles for task design. Task design in mathematics education: An ICMI study 22, 19-81.

[4] Czmoch, I., & Pękala, A. (2014). Traditional design versus BIM based design. Procedia Engineering, 91, 210-215.

[5] Svetel, I., Kosić, T., & Pejanović, M. (2018). Digital Vs. traditional design process. In Proceedings of 5th International Academic Conference on Places and Technologies,"Places and Technologies 2018–Keeping up with technologies to adapt cities for future challenges" (pp. 453-460). Belgrade: Faculty of Architecture.

[6] Goto, S., Gianfagia, T. J., Munafo, J. P., Fujii, E., Shen, X., Sun, M., ... & Herrup, K. (2017). The power of traditional design techniques: The effects of viewing a Japanese garden on individuals with cognitive impairment. HERD: Health Environments Research & Design Journal, 10(4), 74-86.

[7] Aktaş, M., Yetgin, Z., Kılıç, F., & Sünbül, Ö. (2022). Automated test design using swarm and evolutionary intelligence algorithms. Expert Systems, 39(4), e12918.

[8] Tzotzis, A., García-Hernández, C., Huertas-Talón, J. L., & Kyratsis, P. (2020). CAD-based automated design of FEA-ready cutting tools. Journal of Manufacturing and Materials Processing, 4(4), 104.

[9] Yang, T., Jin, G. F., & Zhu, J. (2017). Automated design of freeform imaging systems. Light: Science & Applications, 6(10), e17081-e17081.

[10] Appleton, E., Madsen, C., Roehner, N., & Densmore, D. (2017). Design automation in synthetic biology. Cold Spring Harbor perspectives in biology, 9(4), a023978.

[11] Kim, Y. M. (2017). Development of automated structural design tool for horizontal members of Hanok. Journal of the Architectural Institute of Korea Structure & Construction, 33(4), 21-28.

[12] Bianconi, F., Filippucci, M., & Buffi, A. (2019). Automated design and modeling for mass-customized housing. A web-based design space catalog for timber structures. Automation in construction, 103, 13-25.

[13] Otero-Muras, I., Henriques, D., & Banga, J. R. (2016). SYNBADm: a tool for optimization-based automated design of synthetic gene circuits. Bioinformatics, 32(21), 3360-3362.

[14] Jia, R., Jin, B., Jin, M., Zhou, Y., Konstantakopoulos, I. C., Zou, H., ... & Spanos, C. J. (2018). Design automation for smart building systems. Proceedings of the IEEE, 106(9), 1680-1699.

[15] Runge, G., & Raatz, A. (2017). A framework for the automated design and modelling of soft robotic systems. CIRP Annals, 66(1), 9-12.

[16] Liu, B., Gielen, G., & Fernández, F. V. (2014). Automated design of analog and high-frequency circuits. A computational intelligence approach, Springer, Berlin, Heidelberg, 978-3.

[17] Yi, W., Qu, R., & Jiao, L. (2023). Automated algorithm design using proximal policy optimisation with identified features. Expert Systems with Applications, 216, 119461.

[18] Xiaofeng Yuan, Hui Qin, Wei Cao, Taiheng Zhang & Xinqiang Niu. (2025). Optimal Scheduling Study of Hydro–Solar Complementary System Based on Improved Beluga Whale Algorithm. Water,17(6),878-878.

[19] Shichang Zhao, Zhonghao Zhang & Shuai Wang. (2025). Efficient solution method for power facility relocation planning based on SVM-PSO optimization. Results in Engineering,26,104601-104601.

[20] Jisheng Wei, Chunyue Huang, Chao Gao & Gui Wang. (2025). Stress analysis and optimization of QFN solder joints under bending-torsion coupled conditions based on the SSA. Microelectronics Journal,159,106661-106661.

[21] Minan Tang, Yude Jiang, Shuyou Yu, Jiandong Qiu, Hanting Li & Wenxin Sheng. (2025). Non-dominated sorting WOA electric vehicle charging station siting study based on dynamic trip chain. Electric Power Systems Research,244,111532-111532.