symmetry

MDPI

Article

An Improved Multi-Objective Harris Hawk Optimization with Blank Angle Region Enhanced Search

Zhicheng Yan [1], Qibing Jin [1], Yang Zhang [2,*], Zeyu Wang [1] and Ziming Li [1]

[1] College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China; 2019200751@mail.buct.edu.cn (Z.Y.); jinqb@mail.buct.edu.cn (Q.J.); wangzeyu@mail.buct.edu.cn (Z.W.); 2019200767@mail.buct.edu.cn (Z.L.)

[2] College of Mechanical and Electrical Engineering, Beijing University of Chemical Technology, Beijing 100029, China

* Correspondence: 2002500011@mail.buct.edu.cn

Abstract: Aiming at the problems of low precision, low search efficiency, and being easy to fall into local optimization of the multi-objective harris hawk optimization algorithm (MOHHO) based on grid method, a MOHHO based on blank angle region enhanced search (BARESMOHHO) is proposed. The main changes of the algorithm are as follows: firstly, chaotic mapping is used to initialize the population, which is beneficial to speed up the search. Then, in order to find low-density regions faster, the algorithm adjusts the classification level according to the number of individuals in the external archive. In order to make the distribution of individuals in the target space more uniform, inspired by the idea of symmetrical segmentation, the number of archives at different levels are symmetrically distributed. Finally, it strengthens the search for the non-individual region (blank angle region) in the process of division. The effectiveness of the proposed algorithm is verified by comparing it with some known classical functions on test functions.

Keywords: multi-objective optimization; harris hawk optimization; computation intelligence; swarm intelligence

Citation: Yan, Z.; Jin, Q.; Zhang, Y.; Wang, Z.; Li, Z. An Improved Multi-Objective Harris Hawk Optimization with Blank Angle Region Enhanced Search. *Symmetry* **2022**, *14*, 967. https://doi.org/10.3390/sym14050967

Academic Editor: José Carlos R. Alcantud

Received: 16 March 2022
Accepted: 5 May 2022
Published: 9 May 2022

1. Introduction

The multi-objective optimization problem has many applications in real life, such as electrical, machine learning, biological, and the internet [1–5]. Generally speaking, the problems to be dealt with in these fields need to be evaluated by at least two or more goals, but these goals are usually mutually restrictive or even contradictory and cannot be linked with a definite qualitative relationship. It is very difficult to find the optimal solution to make each optimization goal achieve the optimal one simultaneously. Because of the particularity of this kind of problem, the solution of multi-objective optimization is usually not a single one but a set of many solutions. Judging the merits and demerits of each solution becomes a complex problem with multiple objective evaluation criteria. With the increase in target number and target dimension, the difficulty of creating an optimal solution set will be greatly increased.

At present, population-based evolutionary algorithms are widely used in the field of multi-objective optimization. The progress of multi-objective algorithms benefits from the progress of single-objective algorithms and computer technology. The evolutionary algorithm has made more and more achievements in the application of multi-objective optimization. According to the choice of solutions, the current evolutionary algorithms can be divided into three categories:

The first one is indicator-based multi-objective algorithms. In these algorithms, individuals are evaluated by some known or user-defined metrics. Indicator-based multi-objective algorithm (IBEA) [6], first proposed by Zitzler and Künzli, uses predefined optimization goals to measure the contribution of each solution. A multi-objective algorithm (R2-EMOA) based on R2 indicators was proposed by Heike Trautmann et al. [7]. The multi-objective algorithm (QHV) based on the HV index was proposed by L. M. S. Russo and A. P. Francisco [8].

The second category is based on decomposition or aggregation methods. The first classical algorithm (MOEA/D) based on decomposition strategy was proposed by Q. Zhang and H. Li [9]. Some mathematical programming methods such as the weighted sum approach, Tchebycheff approach, and Penalty-based boundary intersection are used to transform multi-objective problems into single-objective problems. A new algorithm (MOEA/D-DE) [10] has been proposed, combining the differential evolution algorithm commonly used to solve single-objective problems with MOEA/D. This algorithm proves that the decomposition-based multi-objective evolutionary algorithm has a good application prospect in dealing with complex shapes of Pareto frontiers. An improved MOEA/D algorithm (MOEA/D-AWA) [11], which can adaptively adjust the weights of single-objective problems, is proposed by Yutao Qi and Xiaoliang Ma to help the non-inferior solution set distribute more uniformly.

The third category is multi-objective algorithms based on the Pareto dominance relationship. These algorithms can produce multiple solutions at a time. Most of the classical multi-objective algorithms are based on Pareto domination. The new algorithms are formed by combining with some other strategies such as elite strategy, crowding distance operator, and external archives. Among them, the classical algorithms include NSGA-II [12], SPEA2 [13], PESA-II [14], M-PAES [15], MOPSO [16], MOABC [17], etc.

The structure of this paper can be divided into the following sections. The first section introduces the review and classification of multi-objective optimization algorithms. The second section introduces the related research of the multi-objective harris hawk algorithm. The third section introduces the structure and principle of the single-objective harris hawk algorithm. The fourth section introduces the proposed improved strategy and the structure of the BARESMOHHO algorithm in detail. In the fifth section, three comparative experiments are designed, and the results are analyzed to prove the effectiveness of the proposed strategy and the superiority of the improved algorithm. The sixth section is a summary of the paper and the prospect of future work.

2. Related Work

The meta-heuristic intelligent algorithm is a general name for a class of algorithms. These algorithms use mathematical modeling of physical laws objectively existing in nature or survival and reproduction patterns of biological populations to deal with optimization problems. The meta-heuristic intelligent algorithm has the advantages of simple structure, wide application range, and randomness, so it can obtain a better optimization effect compared with the traditional optimization algorithm [18]. There are many kinds of meta-heuristic algorithms, such as the genetic algorithm (GA) [19], simulated annealing (SA) [20], ant colony algorithm (ACA) [21], particle swarm optimization (PSO) [22], artificial bee colony (ABC) [23], whale algorithm (WOA) [24], gray wolf algorithm (GWO) [25], harris hawks optimization (HHO) [26], spotted hyena optimizer (SHO) [27], the farmland fertility [28], and so on.

This paper mainly studies the harris hawk optimization, which was first proposed in 2019. It is a meta-heuristic intelligent optimization algorithm to simulate the hunting behavior of the harris hawk. The significant advantages of the HHO are its simplicity and have a few exploratory and exploitative mechanisms. Moreover, it has fewer control parameters and fast convergence speed. Due to its excellent performance, the corresponding multi-objective harris hawk optimization algorithm is also generated, which combines external file policy and roulette rules with the HHO algorithm [29]. PeiDu et al. designed a new roulette probability operator and proposed a new MOHHO algorithm combined with the archive strategy. The algorithm was used to adjust the parameters of the extreme learning Machine (ELM) to achieve high-precision air quality prediction [30]. Wenlong Fu et al. proposed a multi-objective harris hawk algorithm (HMOHHO) to solve the PID parameter tuning of the hydraulic turbine control system. The escape energy model of the HHO algorithm was modified into a nonlinear energy model. In the algorithm, Latin hypercube sampling was used for population initialization to achieve uniform distribution of the initial population in the decision space. In addition, a mutation strategy is introduced to enhance the diversity of the solution set [31]. Ali Selim et al. proposed a multi-objective harris hawk algorithm based on a gray relational strategy to find the compromise solution to solve the problem of finding the optimal distribution mode in the distribution network [32]. Mohammad Asif Hossain et al. applied MOHHO to deal with routing problems in vehicle self-organized networks (VANET) and found in application tests that the optimization effect of this algorithm was superior to the previous algorithm [33]. Jayashree Piri et al. used binary-coded MOHHO (MOQBHHO) to solve feature selection problems and compared it with other multi-objective optimization algorithms for feature selection, proving that MOQBHHO has advantages in solving feature selection problems [34]. Benyamin Abdollahzadeh et al. proposed a hybrid algorithm named MOHHOFOA, which mixed MOHHO and MOFOA. This algorithm has been applied to the feature selection problem of data mining and, compared with other commonly used algorithms, it has proved to be effective [35]. Studies on MOHHO have been applied in many fields, and the research heat of this algorithm is gradually increasing, so MOHHO has profound research significance.

After analyzing previous versions of the multi-objective harris hawk optimization algorithm, most of them are based on the grid selection method. We find that there are many blank regions when using the grid selection method, which will affect the search efficiency of the multi-objective optimization algorithm and may cause the algorithm to fall into local optimal. To remedy these deficiencies, a new multi-objective harris hawk optimization algorithm with blank angle region enhanced search is proposed in this paper. The main contributions of this algorithm are as follows:

(1) The angle segmentation method is introduced into an external archive to divide the target space. An adaptive partition strategy is designed according to the number of non-inferior solutions of external archives.
(2) Blank angle region enhanced search. In the early stage of the algorithm, empty regions may appear in the target space, for which the algorithm is guided to explore the empty region by selecting its neighborhood. The algorithm introduced with this strategy saves calculation time and improves search efficiency.
(3) Chaos strategy is introduced and combined with the proposed algorithm. The Tent chaotic map is selected as the initialization method of the algorithm through experiments. This method improves the search speed of the algorithm.

3. Harris Hawk Optimization Algorithm

The harris hawk optimization algorithm was first proposed in 2019 [26], which is a new bionic intelligent optimization algorithm simulating the characteristics of harris hawk predation. The algorithm contains two stages: the exploration phase and the exploitation phase.

3.1. Exploration Phase

Each harris hawk represents a solution. According to the different situations where the harris hawk does and does not find prey, the updated mathematical model is as follows:

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1|X_{rand}(t) - 2r_2X(t)| & q \geq 5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q < 5 \end{cases} \tag{1}$$

where $X(t+1)$ is the position vector of hawks in the next iteration t, $X(t)$ is the current position vector of hawks, $X_{rabbit}(t)$ is the location of the prey. $X_{rand}(t)$ is a randomly selected hawk from the current population. r_1, r_2, r_3, r_4, and q are random numbers between 0 and 1. UB and LB are the upper and lower limits of the search space, respectively. $X_m(t)$ is the average position of all individuals in the current hawk group. $X_m(t)$ can be described by the following formula:

$$X_m(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \tag{2}$$

where $X_i(t)$ is the location of each hawk in iteration t and N represents the total number of hawks.

The HHO algorithm uses an escape energy model to achieve the transition from exploration to exploitation and varies between different exploitation behaviors. The escape energy model can be described as:

$$E = 2E_0\left(1 - \frac{t}{T}\right) \tag{3}$$

where E is the escape energy of prey, T is the maximum iteration number, and t is the current iteration number. E_0 is the initial value of escape energy. When $|E| \geq 1$, the exploration phase is executed. When $|E| < 1$, the exploitation phase is executed.

3.2. Exploration Phase

The prey has the probability to escape when being chased: λ is set as the probability of prey escape. HHO algorithm switches four simulated hunting behaviors according to λ and E.

Soft besiege: when $\lambda \geq 0.5$ and $|E| \geq 0.5$, the prey has enough energy and escapes the capture by jumping, while the harris hawk will gradually consume the energy of the prey, and then choose the best position to capture the prey. The updated equation is as follows:

$$X(t+1) = \Delta X(t) - E|JX_{rabbit}(t) - X(t)| \tag{4}$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \tag{5}$$

where $\Delta X(t)$ is the difference between the position vectors of hawk and prey. $J = 2(1 - r_5)$ represents the random jump energy of prey in the process of escape. r_5 is random numbers between 0 and 1.

Hard besiege: when $\lambda \geq 0.5$ and $|E| < 0.5$, the prey is exhausted and the escape energy is very low. The harris hawk chooses to attack rapidly. The updated strategy is as follows:

$$X(t+1) = X_{rabbit}(t) - E|\Delta X(t)| \tag{6}$$

Soft besiege with progressive rapid dives: when $\lambda < 0.5$ and $|E| \geq 0.5$, the prey has enough energy to escape. In order to simulate the prey escape mode, levy Flight (*LF*) function is integrated into HHO algorithm. The updated strategy is as follows:

$$X(t+1) = \begin{cases} Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X(t)| & if\ F(Y) < F(X(t)) \\ Z = Y + S \times LF(D) & if\ F(Z) < F(X(t)) \end{cases} \tag{7}$$

where D is the dimension of the problem, and S is the D-dimensional random column vector. The mathematical model of LF function is as follows:

$$LF(x) = 0.01 \times \frac{\mu \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}}\right)^{\frac{1}{\beta}} \tag{8}$$

where μ and v are random numbers between 0 and 1, and β is the default constant 1.5.

Hard besiege with progressive rapid dives: when $\lambda < 0.5$ and $|E| < 0.5$, the prey does not have enough energy to escape. The updated strategy is as follows:

$$X(t+1) = \begin{cases} Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X_m(t)| & if\ F(Y) < F(X(t)) \\ Z = Y + S \times LF(D) & if\ F(Z) < F(X(t)) \end{cases} \tag{9}$$

4. Improved Multi-Objective Harris Hawk Algorithm

After analyzing the previous versions of the multi-objective harris hawk algorithm, most algorithms are inspired by the multi-objective particle swarm optimization algorithm. The external archive maintenance strategy of these algorithms is based on the grid method. We find that the grid method will produce many empty regions when dividing the target space, and more empty regions will waste computing resources. We designed some strategies to improve the algorithm.

4.1. The Strategy of Angle Region Division

The algorithm sets an external archive in advance to store the generated non-inferior solutions. Each iteration of the algorithm produces a set of non-inferior solutions whose fitness value is $f_k^{num_{archive}}$, $(k = 1, 2)$, $num_{archive}$ is the number of external archive individuals. For these non-inferior solutions, we use the following Formula (10) to get their angle information to divide the region and calculate the density.

$$\theta^{num_{archive}} = \arctan\frac{f_1^{num_{archive}}}{f_2^{num_{archive}}} \tag{10}$$

After obtaining their angle information, we use Formula (11) to standardize the fitness values of non-inferior solutions to facilitate the division of the target space.

$$f_{sk}^{num_{archive}} = \frac{f_k^{num_{archive}} - \min(f_k^{num_{archive}})}{\max(f_k^{num_{archive}}) - \min(f_k^{num_{archive}})} \tag{11}$$

At the initial stage of the algorithm, the number of non-inferior solutions generated is small. If the target space is divided by a fixed number of regions, there will be many equal density regions, which is not conducive to the choice of leaders. As shown in Figure 1, the number of non-inferior solutions for external files is small in the early iteration, assuming that we divide the target space from a fixed number of angle regions (here, we set the number of regions to 12 to illustrate). Each region has an individual in region 1–8.

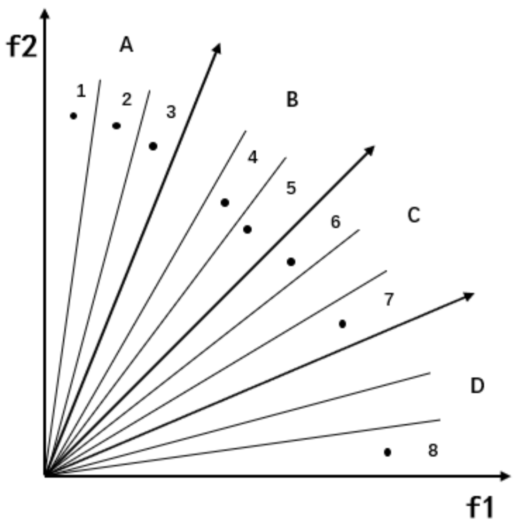


Figure 1. Schematic diagram of angle region division strategy.

According to the rules of the roulette wheel method, when choosing a leader, all eight individuals are equally likely to be selected. If we divide by regions A, B, C, and D, region D is the real region with the lowest density, so we should prefer to choose the solution in region D. Therefore, the number of regions divided by external archives should be adjusted at any time according to the number of individuals in external archives. This method helps to better select low-density regions. We defined five segmentation levels based on the number of individuals stored in the external archive per iteration. The number of archives is inspired by the idea of symmetrical segmentation, which is to divide the region in half. This process can be described by the following mathematical formulas:

$$\frac{i-1}{5}\max_{archive} < num_{archive} \leq \frac{i}{5}\max_{archive}, i = 1,2,3,4,5 \tag{12}$$

$$num_{division} = 4 + 24 \times (i-1) \tag{13}$$

where $\max_{archive}$ is the maximum external file capacity, and $\max_{archive}$ is set to 100 in the algorithm proposed in this paper. i is the classification level, $num_{division}$ is the number of archives corresponding to different classification level.

In this way, in the early stage of the algorithm, the number of external file individuals was small, and the number of divided regions was small so that the real low-density regions could be accurately found. As the number of external file individuals increases, the number of divided regions increases. Ideally, when only one individual exists in each region, iteration will be stopped, or iteration will be stopped when the maximum number of iterations is reached.

4.2. Blank Angle Region Enhanced Search

In the process of dividing the target space, there will be non-individual regions. The reason why there is a non-individual region is that there is no solution, or the region has not been searched. In general, grid methods ignore non-individual regions and select individuals in low-density regions.

In this way, there is a problem. As shown in Figure 2, there are high-density regions 1 and 2 near the non-individual regions, but the probability of selecting this region is very low. The population iteration will be carried out according to the individuals in low-density regions 3 or 4, and the possibility of falling into local optimal is relatively high. If a solution exists in the non-individual region and the search for the region is not strengthened, the search efficiency will be reduced, although the solution may be found in the region in subsequent iterations. In order to deal with the non-individual region, a blank region enhanced search strategy is proposed. In the presence of a non-individual region, the region with individuals is selected in the neighborhood of the non-individual region, and the individual is selected as the guide. Combining this strategy with the angle region division strategy, the distribution of non-individual regions will occur in the following three cases:

Case 1: There is only one blank region in the target space. As shown in Figure 3, region 3 is a non-individual region, and there are individuals in other regions. In this case, the non-individual region is likely to have solutions, so select the low-density region 4 in neighborhoods 2 and 4, and select the individual **a** closer to region 3. For the convenience of later expression, the individual whose angle is closest to the individual-free region (such as individual **a**) is called the candidate solution If two neighborhoods have the same density, randomly select the candidate solution in a region.

Case 2: There are many non-individual regions, and they are all adjacent. As shown in Figure 4, the three non-individual regions of regions 4–6 are adjacent, so these three regions are considered as a region. Although the density in region 3 is higher than that in regions 1 and 2, selecting individuals in region 3 can help explore non-individual regions and select candidate solution b closer to region 3.

Case 3: There are many non-individual regions, and they are distributed at intervals. As shown in Figure 5, regions 1, 3, and 5 are separated by regions 2, 4, and 6; regions 1 and

5 have the same density. Individual c is selected as the candidate solution by comparing the individual information in the two regions. If the density of regions 1, 3, and 5 is different, the individual in the lowest density region is selected as the candidate solution.

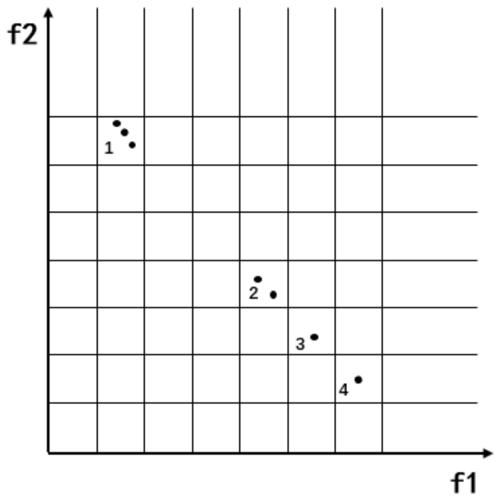


Figure 2. Diagram of individual distribution in grid method.

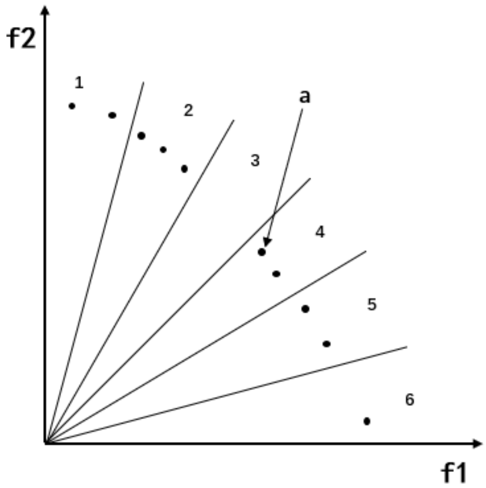


Figure 3. Schematic diagram of Case 1.

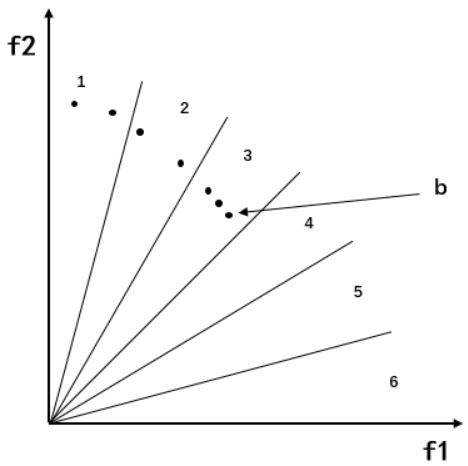


Figure 4. Schematic diagram of Case 2.

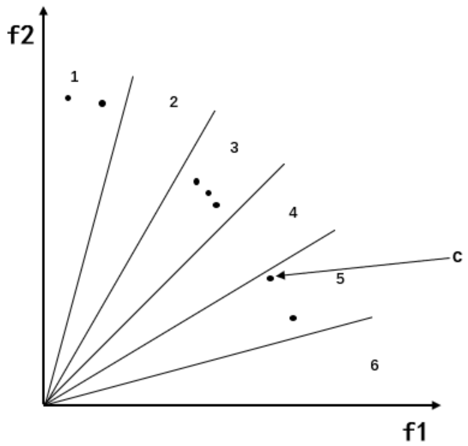


Figure 5. Schematic diagram of Case 3.

The pseudo-code of the blank angle region enhanced search strategy is shown in Algorithm 1.

Algorithm 1 pseudo code of the blank angle region enhanced search strategy

---

Inputs: Values of non-inferior solutions of populations $f_k^{num_{archive}}$, current number of non-inferior solutions $num_{archive}$.
  Obtain individual angle information through Formula (10) and standardize using Formula (11).
if $num_{archive} < \max_{archive}$ do
  Use Formulas (12) and (13) to determine $num_{division}$. Calculate the number of individuals in the region and obtain the number of non-individual regions.
if number of non-individual regions ==0
  Execute the roulette wheel to choose the leader.
end
if number of non-individual regions ==1 **then**
  Case 1;
else if number of non-individual regions >1 and adjacent **then**
  Case 2;
else if number of non-individual regions >1 and non-adjacent **then**
  Case 3;
else
  Remove excess individuals from high density regions.
Output: Selected individual leader $X_{rabbit}(t)$

4.3. Initialize the Population Using Chaotic Map

Most of the multi-objective swarm intelligence optimization algorithms adopt random initialization. The chaotic map sequence has the property of ergodicity and non-repetition [36]. The algorithm proposed in this paper uses a tent map to initialize the population, and the initialization formula is as follows: each harris hawk represents a solution. According to the different situations where the harris hawk does and does not find prey, the updated mathematical model is as follows:

$$X = lb + chaos(m \times dim) \times (ub - lb) \tag{14}$$

where m is the number of individuals, dim is the dimension of the problem, ub and lb are the upper and lower limits of the decision variables, respectively.

Combining the harris hawk algorithm with the above strategies, a multi-objective harris hawk algorithm with blank angle region enhanced search is proposed. The pseudo-code of the BARESMOHHO algorithm is shown in Algorithm 2 and the flow chart of BARESMOHHO is shown in Figure 6.

Algorithm 2 pseudo code of BARESMOHHO

Inputs: Number of individuals m, external archive capacity $\max_{archive}$, maximum iteration maximumiteration
T, problem dimension dim, initial value of chaos U_0.
Initialize population using Formula (14), calculating the fitness value of hawks,
add the non-inferior solution to the external archive.
While $t < T$ **do**
Gets the value of non-inferior solution $f_k^{num_{archive}}$, gets the number of non-inferior
solution $num_{archive}$, run Algorithm 1.
for each hawk **do**
Update escape energy E using Formula (3).
if $|E| \geq 1$ **then**
Exploration phase
use Formula (1) to update the position of the hawk.
if $|E| < 1$ **then**
Exploitation phase
if $\lambda \geq 0.5$ and $|E| \geq 0.5$ **then**
Soft besiege, use Formula (4) to update the position of the hawk.
else if $\lambda \geq 0.5$ and $|E| < 0.5$ **then**
Hard besiege, use Formula (6) to update the position of the hawk.
else if $\lambda < 0.5$ and $|E| \geq 0.5$ **then**
Soft besiege with progressive rapid dives, use Formula (7) to update
the position of the hawk.
else if $\lambda < 0.5$ and $|E| < 0.5$ **then**
Hard besiege with progressive rapid dives, use Formula (9) to update
the position of the hawk.
end for
Boundary detection, calculate the fitness values of the updated hawk population.
Add new solutions to external archive, determine the dominant relationship.
Return Updated external archive

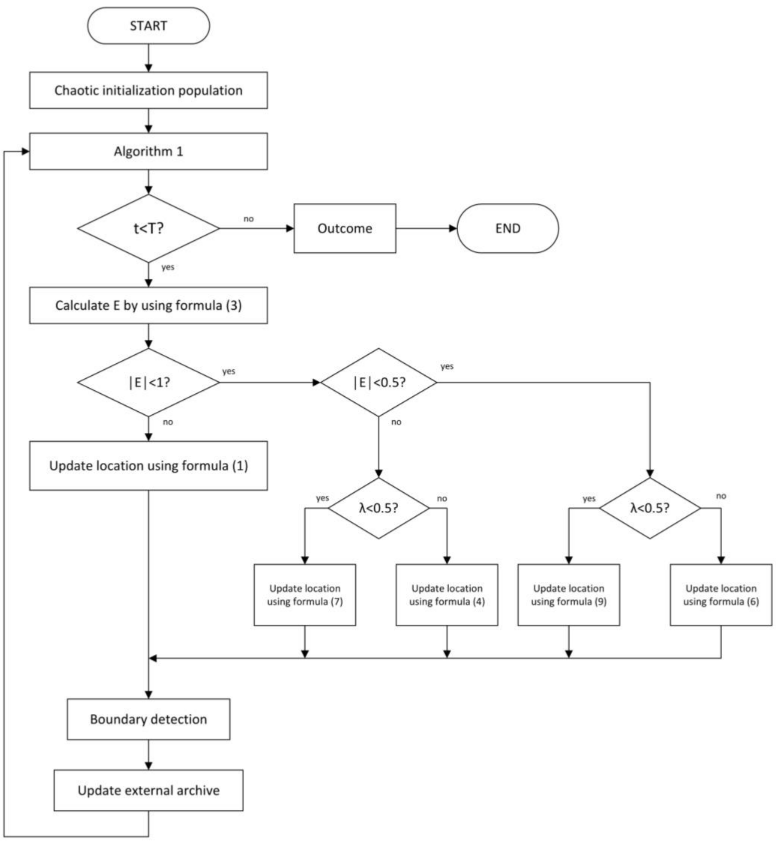


Figure 6. Flow chart of BARESMOHHO.

5. Experimental Results and Discussion

In this chapter, we carry out three experimental verifications of the proposed strategies. We will test them on the ZDT benchmark functions [37] shown in Table 1, where the PF of ZDT1 and ZDT4 functions are convex, the PF of ZDT2 and ZDT6 functions are concave, and the PF of ZDT3 function is disconnected. The function ZDT5 has not been tested because it requires binary encoding. For ZDT1-ZDT3, ZDT4, and ZDT6, 10 decision variables were used in the tests.

Table 1. Description of ZDT test functions.

Function Name	Equation
ZDT 1	$\begin{cases} \min f_1(x_1) = x_1 \\ \min f_2(x) = g(1-\sqrt{f_1/g}) \\ g(x) = 1+9\sum_{i=2}^{m} x_i/(m-1) \\ \text{s.t } 0 \le x_i \le 1, i = 1,2,\ldots,30 \end{cases}$
ZDT 2	$\begin{cases} \min f_1(x_1) = x_1 \\ \min f_2(x) = g\left(1-(f_1/g)^2\right) \\ g(x) = 1+9\sum_{i=2}^{m} x_i/(m-1) \\ \text{s.t } 0 \le x_i \le 1, i = 1,2,\ldots,30 \end{cases}$
ZDT 3	$\begin{cases} \min f_1(x_1) = x_1 \\ \min f_2(x) = g\left(1-\sqrt{(f_1/g)}-(f_1/g)\sin(10\pi f_1)\right) \\ g(x) = 1+9\sum_{i=2}^{m} x_i/(m-1) \\ \text{s.t } 0 \le x_i \le 1, i = 1,2,\ldots,30 \end{cases}$
ZDT 4	$\begin{cases} \min f_1(x_1) = x_1 \\ \min f_2(x) = g\left(1-\sqrt{(f_1/g)}\right) \\ g(x) = 1+10(m-1+\sum_{i=2}^{m} x_i^2 - 10\cos(4\pi x_i)) \\ \text{s.t } 0 \le x_1 \le 1, -5 \le x_i \le 5, i = 2,3,\ldots,10 \end{cases}$
ZDT 6	$\begin{cases} \min f_1(x_1) = 1-\exp(-4x_1)\sin^6(6\pi x_i) \\ \min f_2(x) = g\left(1-(f_1/g)^2\right) \\ g(x) = 1+9\left(\sum_{i=2}^{m} x_i/(m-1)^{0.25}\right) \\ \text{s.t } 0 \le x_i \le 1,\ i = 1,2,\ldots,10 \end{cases}$

In order to measure the performance of the algorithm in the test, we used two performance indicators.

The first indicator is the HV indicator [38]. HV index can evaluate the convergence and distribution performance of the algorithm without relying on real PF. The larger the value of HV, the better the overall performance of the algorithm.

$$HV = \delta\left(\bigcup_{i=1}^{S} v_i\right) \tag{15}$$

S is the non-dominated solution set calculated by the algorithm. v_i represents the volume enclosed by a reference point and a nondominated solution set. $\delta(\bullet)$ is the Lebesgue measure used to evaluate the volume. The reference point coordinates of the HV index in this section are (1,1).

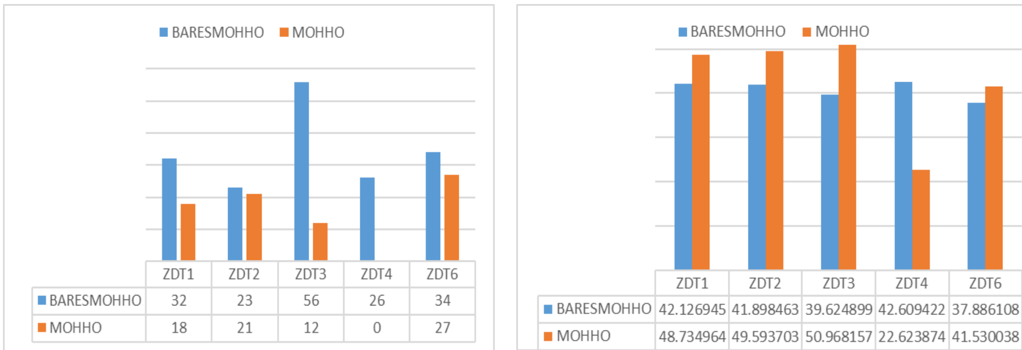
The second indicator is the IGD indicator [39], which is a comprehensive performance evaluation index. It evaluates the convergence and distribution performance of the algorithm by calculating the minimum distance sum between each individual on the real Pareto front and the individual set obtained by the algorithm. The smaller the value of IGD, the better the overall performance of the algorithm.

$$IGD(x, PF) = \frac{\sum \min \mathrm{d}(x, PF)}{|PF|} \tag{16}$$

where min_d(x, PF) denotes the minimum Euclidean distance between x and the points in the reference set PF, x is the solution set calculated by the algorithm, and $|PF|$ is the number of reference points selected in the real Pareto front.

5.1. Experiment 1

The pseudo-code of Algorithm 1 is given in the previous section. In this experiment, Algorithm 1 is compared with the grid method. To ensure the scientific nature of the experiment, the multi-objective harris hawk algorithm based on Algorithm 1 and the multi-objective harris hawk algorithm based on the grid method are both randomly initialized and tested in the same environment. The number of BARESMOHHO and MOHHO populations is set to 200, the external archive capacity is set to 100, the length of the decision variable is 10, and the maximum number of iterations is set to 300. We conducted 10 repeated experiments and chose the smallest results. The results are shown in Figure 7.



(**a**) (**b**)

Figure 7. Experiment 1 results: (**a**) shows the number of iterations when the number of non-inferior solutions for external files reaches maximum capacity; (**b**) shows the running time when the algorithms stop.

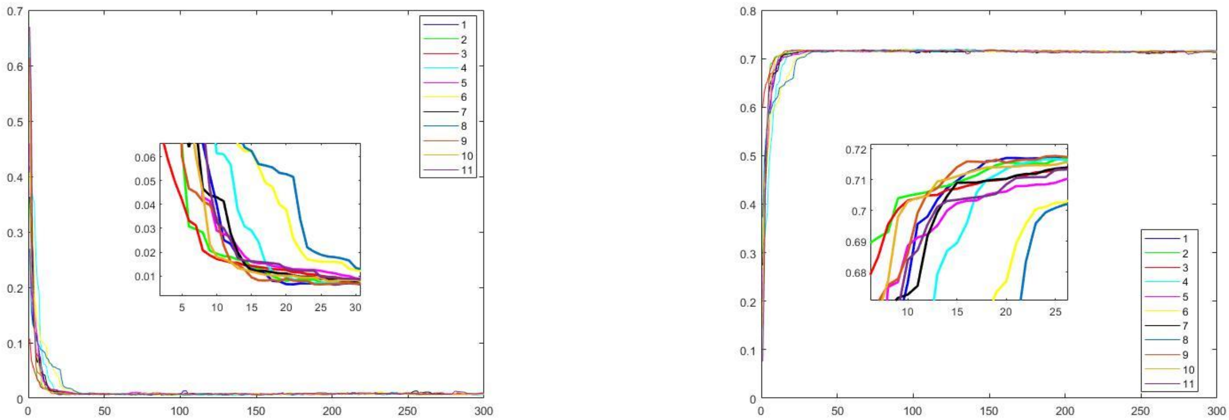
In Figure 7a, ABARESMOHHO has more iterations than MOHHO on five test functions. The result indicates that BARESMOHHO is trying to find solutions in non-individual regions during the search process. This proves the effectiveness of the proposed strategy.

In Figure 7b, the running time of BARESMOHHO is less than that of MOHHO in ZDT1, ZDT2, ZDT3, and ZDT6 test functions. The result shows that the calculation efficiency of BARESMOHHO is higher than that of MOHHO. It should be noted that in the ZDT4 test function, the running time of MOHHO is less than that of BARESMOHHO because MOHHO is trapped in local optimization, which leads to the end of the algorithm soon.

5.2. Experiment 2

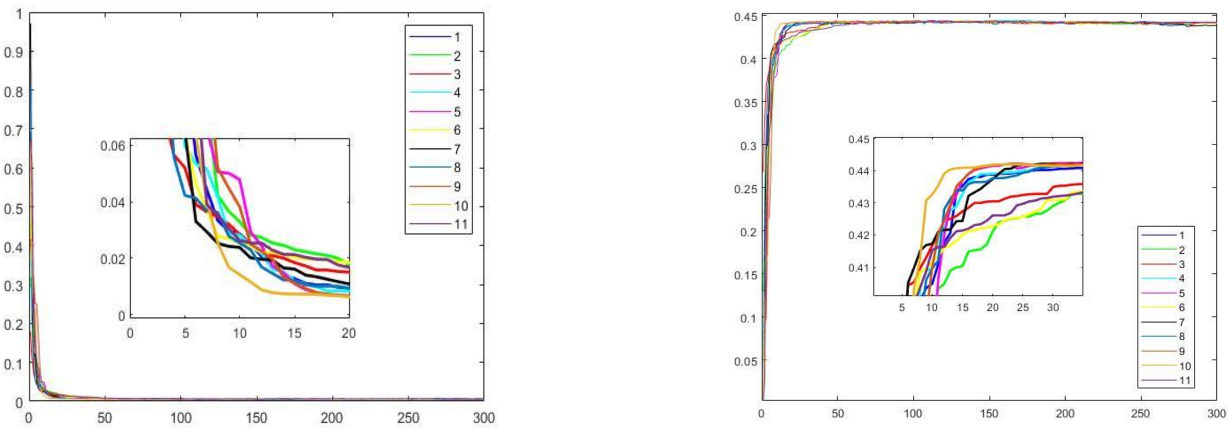
In order to verify the advantages of selecting the tent map as the initialization population, we take 10 chaotic maps as the initialization methods of the algorithm and compare them with the multi-objective harris hawk algorithm based on random initialization. The initial value of the chaotic maps is set to 0.7. The population number of the eleven algorithms is set to 200, the external archive capacity is set to 100, and the maximum number of iterations is set to 300. The dynamic changes of the IGD index and HV index with the number of iterations were taken as references in this experiment, and the results are shown in Figures 8–12. The legends in Figures 8–12 are consistent. The legend 1–10 correspond to the serial numbers of chaotic sequences in Table 2, respectively, and the legend 11 represents the algorithm using random initialization.

Figure 8 shows the IGD and HV dynamic curves. The curve corresponding to legend 9 reaches the flat section of the curve first, which means that this method of legend 9 has the fastest convergence speed in the ZDT1 problem.



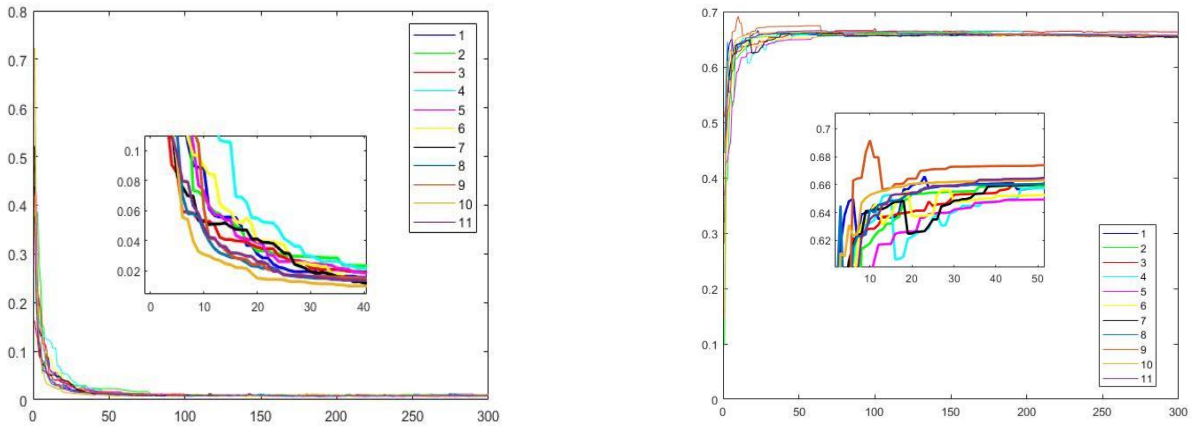
IGD HV

Figure 8. Dynamic change of algorithm index in ZDT 1 with different initialization modes.



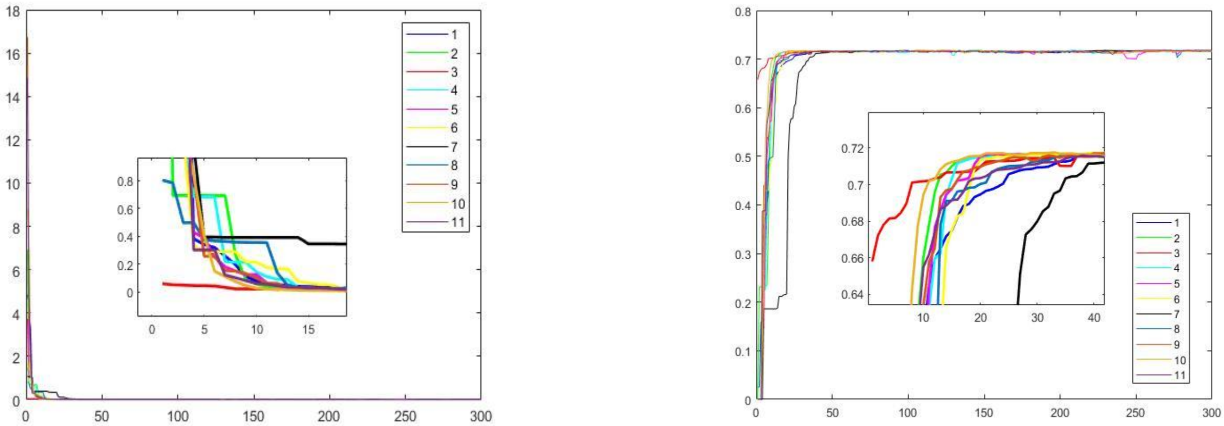
IGD HV

Figure 9. Dynamic change of algorithm index in ZDT 2 with different initialization modes.



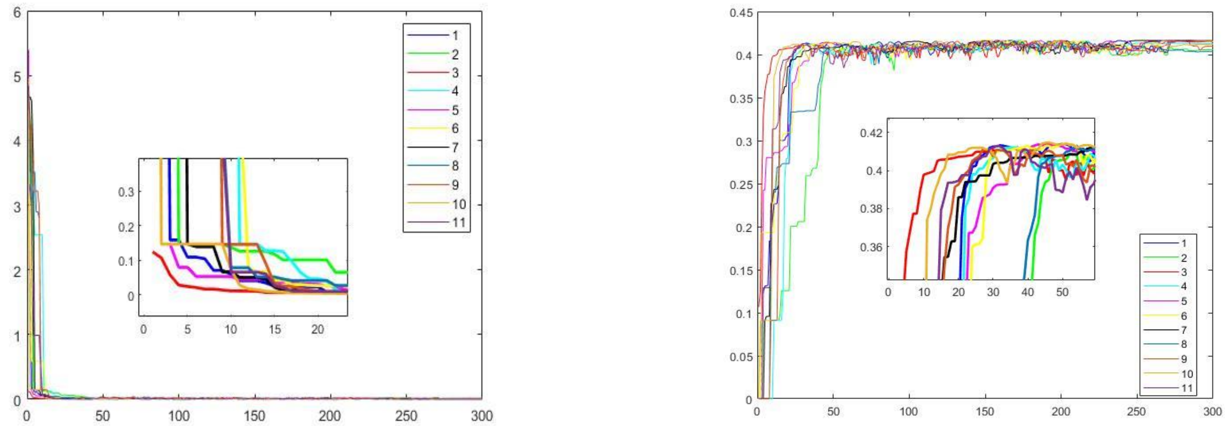
IGD HV

Figure 10. Dynamic change of algorithm index in ZDT 3 with different initialization modes.



IGD HV

Figure 11. Dynamic change of algorithm index in ZDT 4 with different initialization modes.



IGD HV

Figure 12. Dynamic change of algorithm index in ZDT 6 with different initialization modes.

As can be seen from IGD and HV dynamic curves in Figure 9, the curve corresponding to legend 10 is significantly better than other curves, indicating that this method has the fastest convergence speed in the ZDT2 problem.

As can be seen from the IGD and HV dynamic curves in Figure 10, the convergence rate of the legend 10 curve is significantly better than other curves in the IGD curve. In the HV curve, it can be seen that the curve of legend 9 first rises to a high point and then flattens out. This is because the algorithm briefly falls into the local optimum in the early stage and gets a large HV value. Then the algorithm jumps out of the local optimum, and the change of HV value tends to flatten out, indicating that the initialization mode of legend 9 may cause the algorithm to fall into the local optimum. It also shows that the algorithm has the ability to jump out of the local optimum. Except for the curve of legend 9, the convergence rate of the legend 10 curve is significantly better than other curves in the HV curve.

It can be seen from the IGD and HV dynamic curves in Figure 11 that the two index change curves of legend 10 both show the characteristics of rapid convergence, proving that the effect of this initialization method is superior. It should be noted that the curve of legend 3 seems to have a better performance, but its initial solution position vector distribution is too concentrated, so it is not considered.

Table 2. Description of chaotic maps.

No.	Map Name	Map Equation
1	Chebyshev map	$U_{n+1} = \cos\left(\text{n} * \cos^{-1} U_n\right)$
2	Circle map	$U_{n+1} = U_n + b - \left(\frac{a}{2\pi}\right)\sin(2\pi U_n)\text{mod}(1),\ a = 0.5$ and $b = 0.2$
3	Gauss map	$U_{n+1} = \begin{cases} 0 & U_n = 0 \\ \frac{1}{U_n mod(1)} & \text{otherwise} \end{cases}$
4	Iterative map	$U_{n+1} = \sin\left(\frac{a\pi}{U_n}\right),\ a = 0.7$
5	Logistic map	$U_{n+1} = aU_n(1 - U_n),\ a = 4$
6	Piecewise map	$U_{n+1} = \begin{cases} \frac{U_n}{P} & 0 \le U_n < P \\ \frac{U_n - P}{0.5 - P} & P \le U_n < \frac{1}{2} \\ \frac{1 - P - U_n}{0.5 - P} & \frac{1}{2} \le U_n < 1 - P \\ \frac{1 - U_n}{P} & 1 - P \le U_n < 1 \end{cases},P = 0.4$
7	Sine map	$U_{n+1} = \frac{a}{4}\sin(\pi U_n),\ a = 4$
8	Singer map	$U_{n+1} = a\left(7.86U_n - 23.31U_n^2 + 28.75U_n^3 - 13.302875U_n^4\right), a = 1.07$
9	Sinusoidal map	$U_{n+1} = aU_n^2\sin(\pi U_n)$
10	Tent map	$U_{n+1} = \begin{cases} \frac{U_n}{\alpha} & U_n < \alpha \\ \frac{(1 - U_n)}{1 - \alpha} & U_n \ge \alpha \end{cases},\alpha = 0.7$

It can be seen from the IGD and HV dynamic curves in Figure 12 that the two index change curves of legend 10 have a fast convergence rate. The initial solution vector positions of curves of legends 3 and 5 are too concentrated to be considered. It should be noted that the density of the real optimal solution set of the ZDT6 test function is not uniform, so the middle of the HV curve presents unstable changes.

After analyzing the results in Figures 8–12, we find that although the initialization mode of the tent chaotic map corresponding to legend 10 is slightly worse than that of the Sinusoidal chaotic map corresponding to legend 9 in the ZDT1 test function, it achieves better results in the remaining test functions. Therefore, the tent chaotic map is selected as the initialization method of the BARESMOHHO algorithm.

5.3. Experiment 3

In this experiment, the BARESMOHO proposed in this paper is compared with two classical multi-objective algorithms MOPSO and NSGA-II, as well as MOHHO and MOABC. Among them, MOPSO, NSGA-II, and MOABC have been proved to be practical algorithms in many scenarios. MOHHO is a common version in existing research, so it is used for comparison to prove the effectiveness of the proposed algorithm. The population number of these 5 algorithms is set to 200, the maximum number of iterations is set to 300, and the dimension of the decision variable is set to 10. The external archive capacity of BARESMOHHO, MOPSO, MOHHO, and MOABC is set to 100. In MOPSO, the inflation parameter setting is set to 0.1, the number of grids per dimension is set to 10, leader selection pressure and deletion selection pressure are set to 2, the mutation rate is set to 0.1, inertia weight is set to 0.73, inertia weight damping rate is set to 0.99, and personal learning coefficient and global learning coefficient are set to 2. In NSGA-II, the crossover rate is 0.9, the mutation rate is set to 0.1, and the distribution indexes for SBX and polynomial mutation operators are both set to 20.

The comparison between the PF calculated by the five algorithms and the real PF is shown in Figures 13–17.

As can be seen from Figure 13, the optimal solution sets of the five algorithms converge to the real frontier. BARESMOHHO algorithm's optimal solution set has the most uniform distribution, while the distribution of the other four algorithms has obvious intervals.

It can be seen from Figure 14 that the optimal solution sets of the five algorithms converge to the real frontier. The optimal solution sets of the proposed BARESMOHHO algorithm and NSGA-II algorithm are evenly distributed, while the optimal solution sets of the other three algorithms are not well distributed on the real frontier.

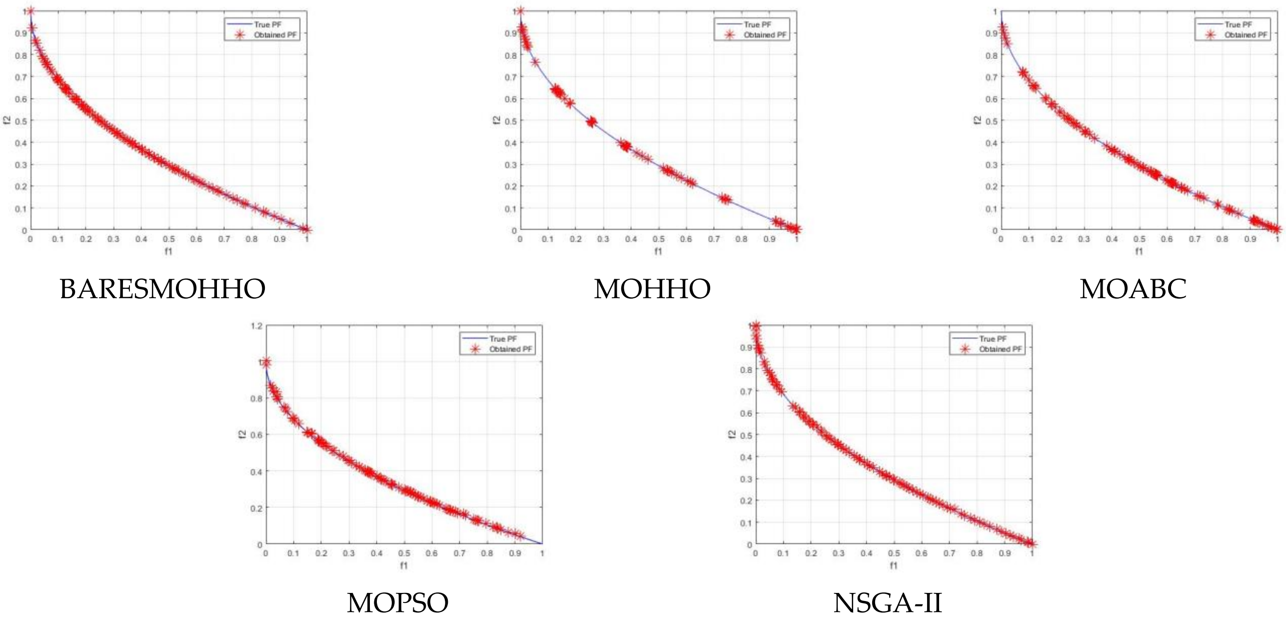


Figure 13. Comparison between PF obtained by five algorithms and real PF on the ZDT 1 test function.

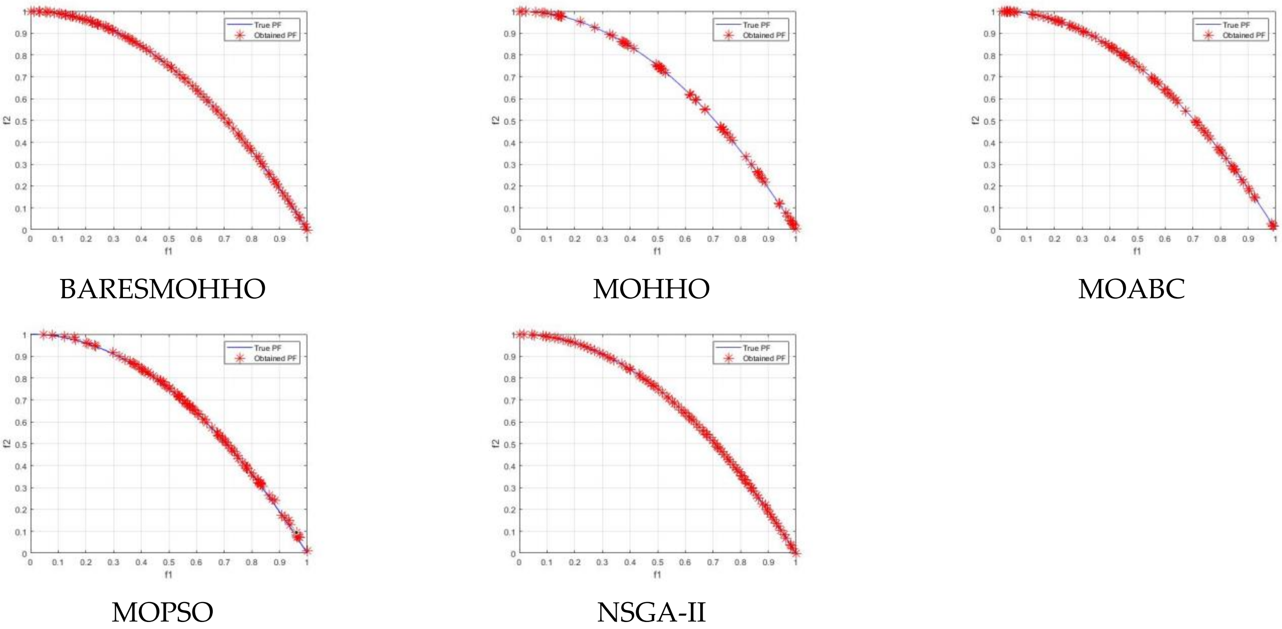


Figure 14. Comparison between PF obtained by five algorithms and real PF on the ZDT 2 test function.

As can be seen from Figure 15, except for MOPSO's optimal solution set, the optimal solution sets of the other four algorithms converge to the real frontier, and the optimal solution sets of the BARESMOHHO algorithm and NSGA-II algorithm are more evenly distributed than the MOHHO algorithm and MOABC algorithm.

As can be seen from Figure 16, only the optimal solution set of the BARESMOHHO algorithm and NSGA-II algorithm converges to the real frontier, while the MOHHO algorithm falls into the local optimum. MOABC has poor search performance on the test function on ZDT 4. The optimal solution set of the MOPSO algorithm is evenly distributed but does not converge to the optimal frontier, and the convergence is poor.

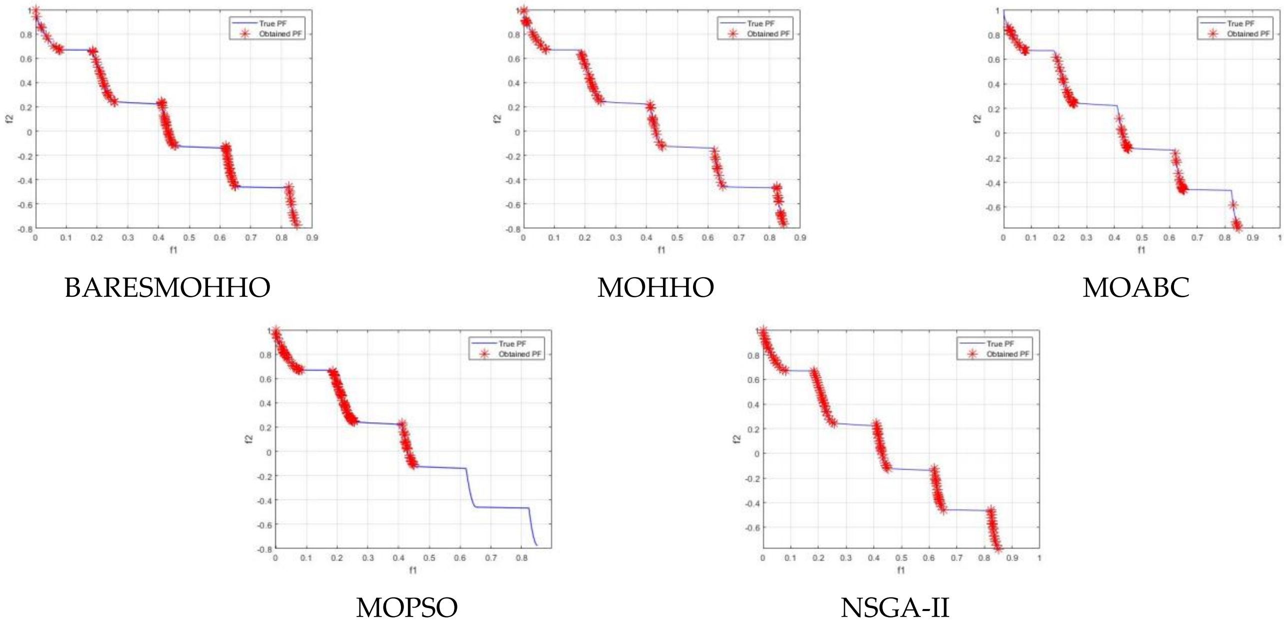


Figure 15. Comparison between PF obtained by five algorithms and real PF on the ZDT 3 test function.

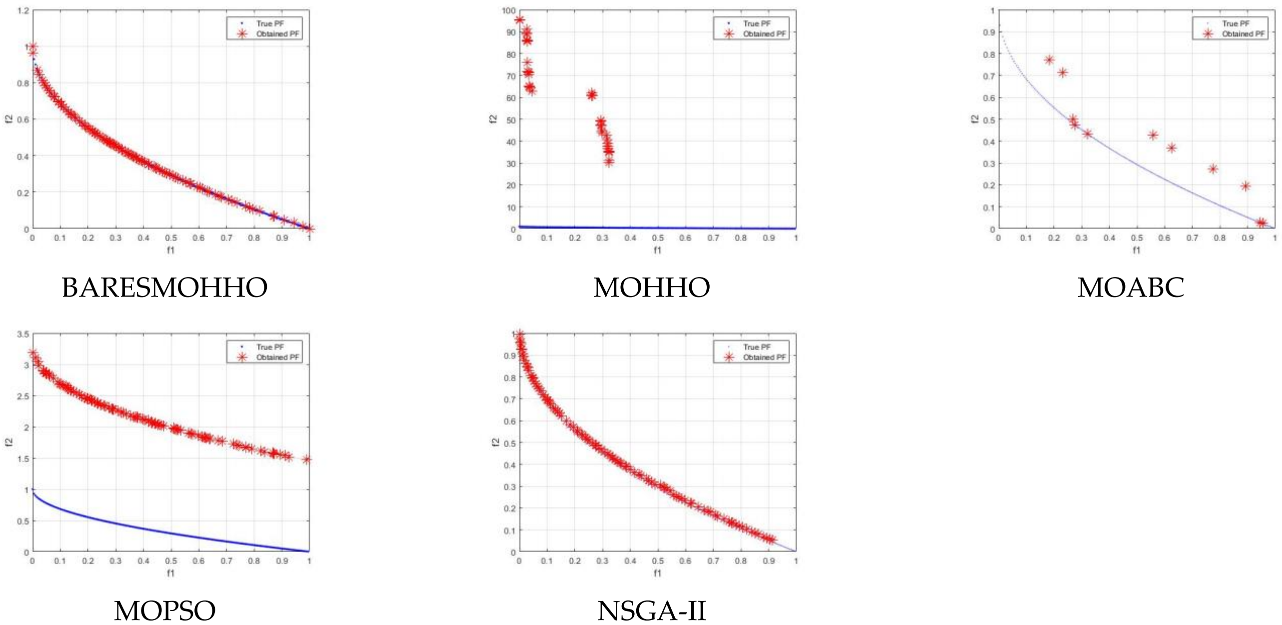


Figure 16. Comparison between PF obtained by five algorithms and real PF on the ZDT 4 test function.

As can be seen from Figure 17, only the optimal solution set of the BARESMOHHO algorithm and NSGA-II algorithm completely converge to the optimal frontier and have good distribution.

Based on the above analysis of the results in Figures 13–17, we find that the optimal solution set calculated by the BARESMOHHO algorithm and NSGA-II algorithm achieves a better distribution result for the distribution of the five test functions.

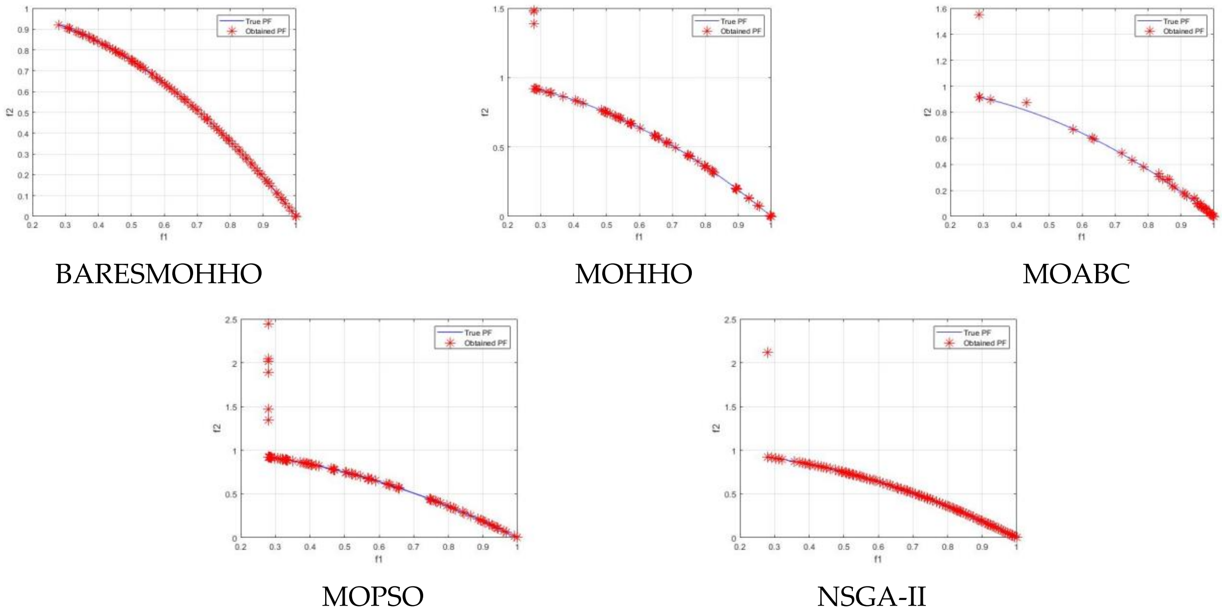


Figure 17. Comparison between PF obtained by five algorithms and real PF on the ZDT 6 test function.

This section also compares the dynamic changes of the HV and IGD of these algorithms on different test functions, and the results are shown in Figures 18–22.

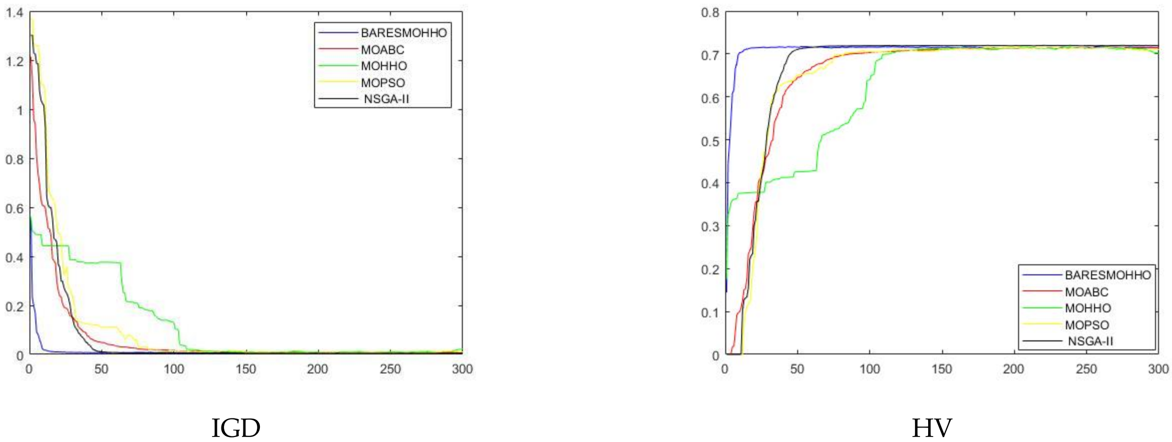


Figure 18. Comparison of dynamic curves of five algorithms on ZDT 1.

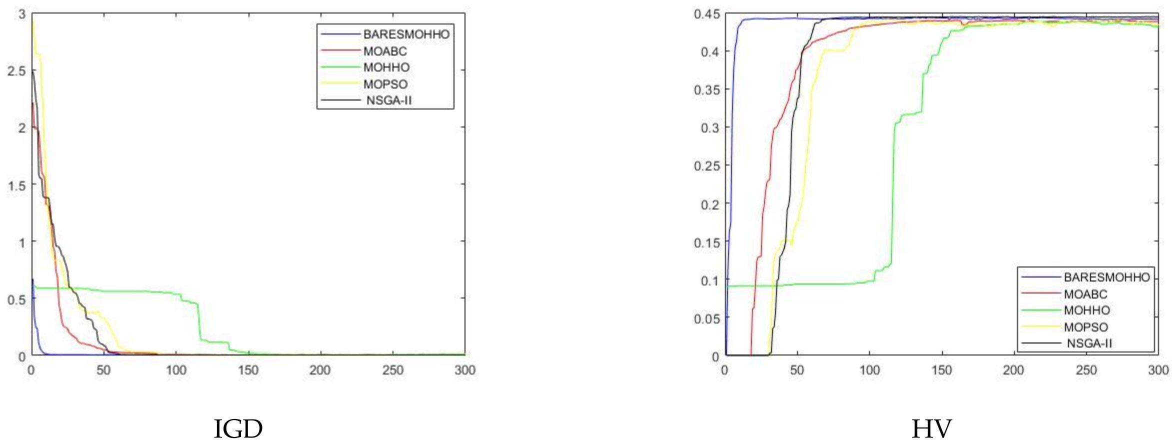
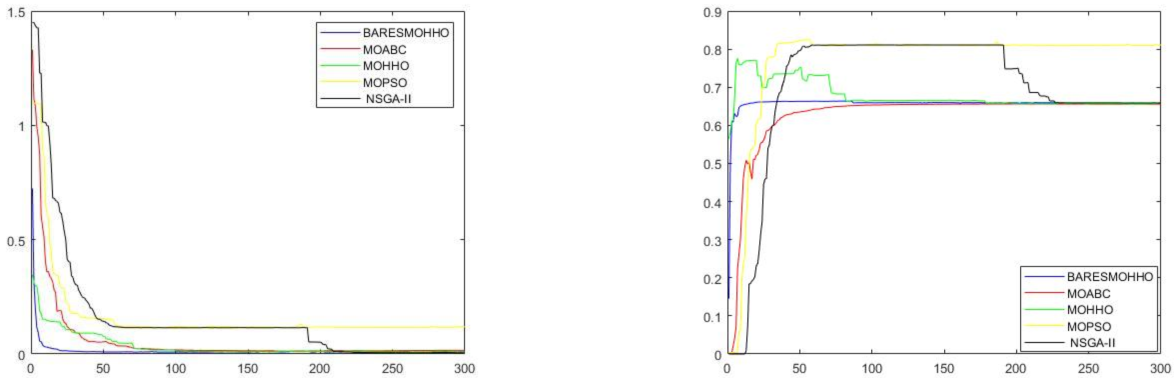
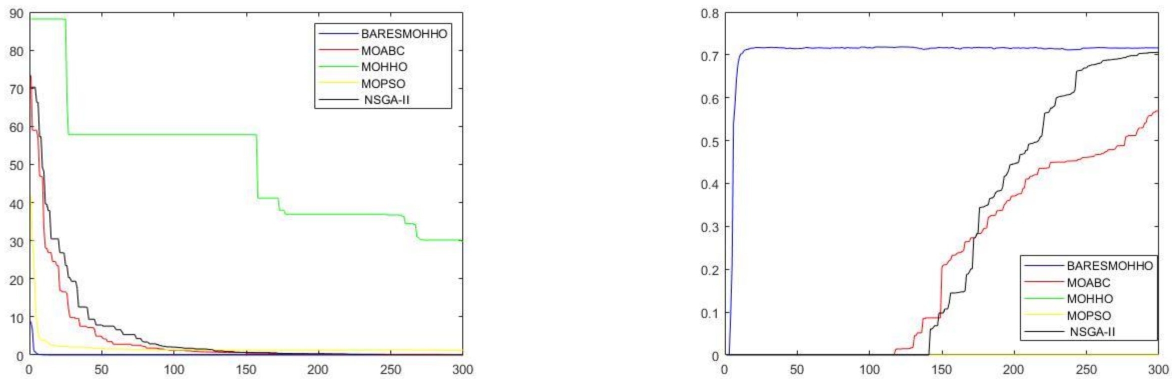


Figure 19. Comparison of dynamic curves of five algorithms on ZDT 2.



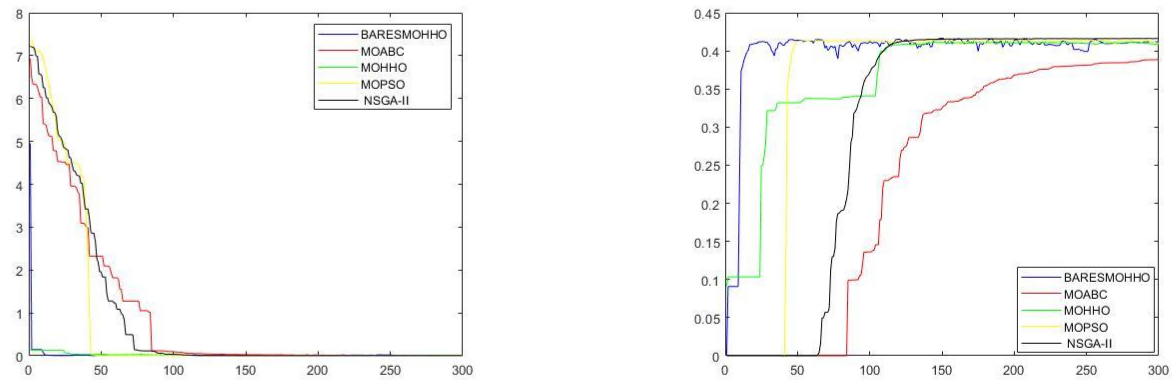
IGD HV

Figure 20. Comparison of dynamic curves of five algorithms on ZDT 3.



IGD HV

Figure 21. Comparison of dynamic curves of five algorithms on ZDT 4.



IGD HV

Figure 22. Comparison of dynamic curves of five algorithms on ZDT 6.

As can be seen from Figure 18, the curve of the BARESMOHHO algorithm converges within dozens of generations, and the other four algorithms converge after 50 generations. It can be seen that the BARESMOHHO algorithm converges faster in the ZDT1 function than other algorithms.

As can be seen from Figure 19, the convergence speed of the two indicators of the BARESMOHHO algorithm is better than that of the other four algorithms. The two indexes of the MOHHO algorithm have initial values in the initial stage of the algorithm, but in the late stage of the algorithm, the HV index has obvious fluctuation, and the MOHHO algorithm runs stably on the ZDT 2 test function. This shows that the BARESMOHHO algorithm not only has fast convergence speed but also runs stably in dealing with non-convex problems.

As can be seen from Figure 20, the convergence speed of the two indicators of the BARESMOHHO algorithm is better than that of other algorithms. In the change of HV index, the change curve of the BARESMOHHO algorithm and MOABC algorithm rises steadily and finally converges. MOHHO and NSGA-II have an unstable stage before convergence. It may be caused by falling into local optimum, which jumps out of local optimum after several iterations, while the MOPSO algorithm falls into local optimum and does not jump out of local optimum in the limited number of iterations. This shows that the BARESMOHHO algorithm performs better than other algorithms when dealing with discontinuous problems.

As can be seen from Figure 21, the convergence speed of the two indicators of the BARESMOHHO algorithm is better than that of other algorithms. In the early stage of the algorithm, the initial IGD value of MOABC and NSGA-II is above 60, while its HV value is 0. This is because the solution calculated by the algorithm is outside the reference point. After the multi-generation calculation, MOABC has not fully received the real frontier, NSGA-II converges to the frontier, and MOPSO has not yet converged to the optimal frontier. It can be considered that the BARESMOHHO algorithm is superior to other algorithms in handling multimodal problems.

As can be seen from Figure 22, the convergence speed of the BARESMOHHO algorithm is better than other algorithms, while the convergence speed of the other four algorithms is slow.

Based on the above analysis of the results in Figures 18–22, we find that the BARESMOHHO algorithm has a fast convergence speed in handling the five types of test functions. This proves that the convergence speed of the BARESMOHHO algorithm is improved compared with other comparison algorithms.

For each test function, run 30 times independently and then statistically analyze the calculation results; the HV and IGD statistical results of different algorithms evaluated on each function are shown in Tables 3 and 4, respectively. The maximum values of HV and minimum values of IGD are bolded for observation.

Table 3. HV statistical results of different algorithms.

		MOABC	**MOHHO**	**MOPSO**	**NSGA-II**	**BARESMOHHO**
ZDT 1	Max	0.7177	0.7160	0.7172	0.7182	**0.7193**
	Min	0.6811	0.4525	0.7034	**0.7133**	0.7126
	Mean	0.7133	0.6974	0.7127	**0.7165**	0.7164
	Std	0.0052	0.0477	0.0018	**0.0016**	**0.0016**
ZDT 2	Max	0.4422	0.4400	0.4426	**0.4447**	0.4446
	Min	0.4343	0.0909	0.4203	0.4432	**0.4433**
	Mean	0.4387	0.1489	0.4366	**0.4442**	0.4439
	Std	0.0017	0.1320	0.0028	**0.0002**	0.0004
ZDT 3	Max	0.6581	0.6925	**0.8161**	0.6595	0.6651
	Min	0.6530	0.3072	**0.6813**	0.6583	0.6522
	Mean	0.6562	0.5839	**0.7519**	0.6588	0.6581
	Std	0.0011	0.1280	0.0574	**0.0001**	0.0021
ZDT 4	Max	0.6412	0.0000	0.0000	0.6974	**0.7181**
	Min	0.3504	0.0000	0.0000	0.6753	**0.6997**
	Mean	0.4957	0.0000	0.0000	0.6851	**0.7169**
	Std	0.0604	0.0000	0.0000	0.0057	**0.0022**
ZDT 6	Max	0.4022	0.4137	0.4148	**0.4165**	**0.4165**
	Min	0.3451	0.0043	0.4056	0.4112	**0.4141**
	Mean	0.3815	0.3547	0.4126	0.4159	**0.4159**
	Std	0.0110	0.1196	0.0010	0.0008	**0.0007**

Table 4. IGD statistical results of different algorithms.

		MOABC	MOHHO	MOPSO	NSGA-II	BARESMOHHO
ZDT 1	Max	0.0109	0.2662	0.0198	0.0175	**0.0093**
	Min	0.0060	0.0082	0.0060	**0.0043**	0.0050
	Mean	0.0071	0.0251	0.0084	**0.0051**	0.0070
	Std	**0.0010**	0.0464	0.0015	0.0015	**0.0010**
ZDT 2	Max	0.0113	1.2752	0.0256	0.0064	**0.0063**
	Min	0.0057	0.0053	0.0063	**0.0043**	0.0056
	Mean	0.0070	1.0637	0.0088	**0.0048**	0.0059
	Std	0.0006	0.4809	0.0014	0.0003	**0.0002**
ZDT 3	Max	0.0282	0.5942	0.2728	0.0076	**0.0071**
	Min	0.0097	0.0085	0.0378	0.0065	**0.0064**
	Mean	0.0153	0.1605	0.1854	**0.0067**	0.0069
	Std	0.0037	0.2301	0.0768	0.0003	**0.0002**
ZDT 4	Max	0.2868	68.4069	1.8721	0.0070	**0.0066**
	Min	0.0569	20.6531	1.1516	0.0052	**0.0051**
	Mean	0.1621	47.4865	1.3256	0.0059	**0.0059**
	Std	0.0433	12.5340	0.1189	0.0005	**0.0003**
ZDT 6	Max	0.0120	0.1363	0.0087	0.0048	**0.0045**
	Min	0.0044	0.0032	0.0026	**0.0022**	0.0033
	Mean	0.0077	0.0206	0.0047	**0.0028**	0.0038
	Std	0.0015	0.0368	0.0010	0.0004	**0.0003**

Table 3 shows the HV numerical results of five algorithms on five test functions. The optimal data (HV minimum) in each row is highlighted in bold in the table. In the test data of the five test functions, the BARESMOHHO algorithm and NSGA-II algorithm contain the same number of optimal values. MOPSO showed a large HV value on the ZDT3 test function, and its value was marked as the optimal value, not because of the superior performance of the algorithm, but because the algorithm fell into a local optimal, and the optimal solution set did not completely converge to the real frontier. Excluding this group of abnormal data, the BARESMOHHO algorithm and NSGA-II algorithm show little difference in performance on the ZDT3 test function. However, in the ZDT4 test function, the BARESMOHHO algorithm outperforms the NSGA-II algorithm in all values.

Table 4 shows the IGD numerical results of five algorithms on five test functions. The optimal data (IGD minimum) in each row is highlighted in bold in the table. From the horizontal view, among the test data of the five test functions, the BARESMOHHO algorithm contains a large number of optimal values, so it can be considered that the BARESMOHHO algorithm has better performance than the NSGA-II algorithm in terms of index values. In the ZDT4 test function, the IGD value of the MOHHO algorithm is too large, and the algorithm falls into the local optimum.

Based on the above analysis of the results in Tables 3 and 4, we believe that the proposed BARESMOHHO algorithm performs better than other algorithms on the five test functions.

6. Conclusions and Future Work

We propose a BARESMOHHO algorithm to extend the HHO algorithm to deal with multi-objective problems. The BARESMOHHO algorithm can easily select real leaders by introducing the angle region division method. In order to improve the search efficiency and obtain the optimal solution set with uniform distribution and good coverage, we adjust the selection method by introducing the blank region enhanced search method. Experiment 1 shows that the blank angle region strategy is effective. In addition, in order to improve the search speed of the algorithm, a tent chaotic map is selected as the initialization method of the algorithm through Experiment 2. Through Experiment 3, on the distribution of the optimal solution set, the BARESMOHHO algorithm and NSGA-II algorithm have better distribution on the five test functions, which shows that these two algorithms can deal with these five kinds of problems well. The BARESMOHHO algorithm has obvious advantages in the convergence speed of the five algorithms, which proves that the rapidity of the improved algorithm is improved. The proposed BARESMOHHO algorithm has certain advantages in the numerical evaluation index. Under comprehensive consideration, the proposed algorithm is superior to several other algorithms of comprehensive performance.

The angle region partition strategy and the blank region processing strategy used in this paper provide a new idea for the improvement of the multi-objective optimization algorithm. In future research, this strategy can be combined with other algorithms to explore whether it has better results. However, this algorithm has certain limitations, and the current research can only deal with the dual-objective optimization problem. Three target versions of this approach can be studied in the future. In this experiment, the algorithm with extensive application research is selected for comparison, which is hoped to prove that the proposed algorithm has the potential to deal with multi-field problems. In future research, the processing effect of the algorithm in specific applications can be explored.

Author Contributions: Conceptualization, Z.Y. and Q.J.; methodology, Y.Z.; software, Z.W.; validation, Z.Y., Q.J. and Y.Z.; formal analysis, Z.L.; investigation, Z.Y.; resources, Z.Y.; data curation, Z.Y. and Q.J.; writing—original draft preparation, Z.Y.; writing—review and editing, Q.J. and Y.Z.; visualization, Y.Z.; supervision, Q.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: In the paper, all the data generation information has been given in detail in the related chapter.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Abido, M.A. A niched Pareto genetic algorithm for multiobjective environmental/economic dispatch. *Int. J. Electr. Power Energy Syst.* **2003**, *25*, 97–105. [CrossRef]
2. Abido, M.A. Multiobjective evolutionary algorithms for electric power dispatch problem. *IEEE Trans. Evol. Comput.* **2006**, *10*, 315–329. [CrossRef]
3. Feng, G.; Lan, Y.; Zhang, X. Dynamic adjustment of hidden node parameters for extreme learning machine. *IEEE Trans. Cybern.* **2014**, *45*, 279–288. [CrossRef] [PubMed]
4. Mnasri, S.; Nasri, N.; van den Bossche, A.; Thierry, V.A. 3D indoor redeployment in IoT collection networks: A real prototyping using a hybrid PI-NSGA-III-VF. In Proceedings of the 14th International Wireless Communications & Mobile Computing Conference (IWCMC), Limassol, Cyprus, 25–29 June 2018; pp. 780–785.
5. Li, X.; Zhang, S.; Wong, K.C. Single-cell RNA-seq interpretations using evolutionary multiobjective ensemble pruning. *Bioinformatics* **2019**, *35*, 2809–2817. [CrossRef] [PubMed]
6. Zitzler, E.; Künzli, S. *Indicator-Based Selection in Multiobjective Search. International Conference on Parallel Problem Solving from Nature*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 832–842.
7. Trautmann, H.; Wagner, T.; Brockhoff, D. *R2-EMOA: Focused Multiobjective Search Using R2-Indicator-Based Selection. International Conference on Learning and Intelligent Optimization*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 70–74.
8. Russo, L.M.S.; Franciscoa, P. Quick hypervolume. *IEEE Trans. Evol. Comput.* **2013**, *18*, 481–502. [CrossRef]
9. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [CrossRef]
10. Li, H.; Zhang, Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans. Evol. Comput.* **2008**, *13*, 284–302. [CrossRef]
11. Qi, Y.; Ma, X.; Liu, F.; Jiao, L.; Sun, J.; Wu, J. MOEA/D with adaptive weight adjustment. *Evol. Comput.* **2014**, *22*, 231–264. [CrossRef]
12. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T.A. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
13. Zitzler, E.; Laumanns, M.; Thiele, L. SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-Rep.* **2001**, *103*. [CrossRef]
14. Corne, D.W.; Jerram, N.R.; Knowles, J.D.; Oates, M.J. PESA-II: Region-based selection in evolutionary multiobjective optimization. In Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, San Francisco, CA, USA, 7 July 2001; pp. 283–290.

15. Knowles, J.D.; Corne, D.W. M-PAES: A memetic algorithm for multiobjective optimization. In Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512), La Jolla, CA, USA, 16–19 July 2000; Volume 1, pp. 325–332.
16. Coello, C.C.; Lechuga, M.S. MOPSO: A proposal for multiple objective particle swarm optimization. In Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), Honolulu, HI, USA, 12–17 May 2002; Volume 2, pp. 1051–1056.
17. Akbari, R.; Hedayatzadeh, R.; Ziarati, K.; Hassanizadeh, B. A multi-objective artificial bee colony algorithm. *Swarm Evol. Comput.* **2012**, *2*, 39–52. [CrossRef]
18. Beheshti, Z.; Shamsuddin, S.M.H. A review of population-based meta-heuristic algorithms. *Int. J. Adv. Soft Comput. Appl.* **2013**, *5*, 1–35.
19. Holland, J.H. Genetic algorithms. *Sci. Am.* **1992**, *267*, 66–73. [CrossRef]
20. Bertsimas, D.; Tsitsiklis, J. Simulated annealing. *Stat. Sci.* **1993**, *8*, 10–15. [CrossRef]
21. Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. [CrossRef]
22. Eberhart, R.; Kennedy, J. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948.
23. Karaboga, D.; Basturk, B. On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.* **2007**, *8*, 687–697. [CrossRef]
24. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
25. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
26. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris Hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]
27. Ghafori, S.; Gharehchopogh, F.S. Advances in spotted hyena optimizer: A comprehensive survey. *Arch. Comput. Methods Eng.* **2021**, 1–22. [CrossRef]
28. Shayanfar, H.; Gharehchopogh, F.S. Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems. *Appl. Soft Comput.* **2018**, *71*, 728–746. [CrossRef]
29. Yüzgeç, U.; Kusoglu, M. Multi-objective Harris Hawks optimizer for multiobjective optimization problems. *BSEU J. Eng. Res. Technol.* **2020**, *1*, 31–41.
30. Du, P.; Wang, J.; Hao, Y.; Niu, T.; Yang, W. A novel hybrid model based on multi-objective Harris Hawks optimization algorithm for daily PM2. 5 and PM10 forecasting. *Appl. Soft Comput.* **2020**, *96*, 106620. [CrossRef]
31. Fu, W.; Lu, Q.P. Multiobjective optimal control of FOPID controller for hydraulic turbine governing systems based on reinforced multiobjective Harris Hawks optimization coupling with hybrid strategies. *Complexity* **2020**, *2020*, 9274980. [CrossRef]
32. Selim, A.; Kamel, S.; Alghamdi, A.S.; Jurado, F. Optimal placement of DGs in distribution system using an improved Harris Hawks optimizer based on single-and multi-objective approaches. *IEEE Access* **2020**, *8*, 52815–52829. [CrossRef]
33. Hossain, M.A.; Noor, R.M.; Yau, K.L.; Azzuhri, S.R.; Z'Abar, M.R.; Ahmedy, I.; Jabbarpour, M.R. Multi-objective Harris Hawks optimization algorithm based 2-Hop routing algorithm for CR-VANET. *IEEE Access* **2021**, *9*, 58230–58242. [CrossRef]
34. Piri, J.; Mohapatra, P. An analytical study of modified multi-objective Harris Hawk Optimizer towards medical data feature selection. *Comput. Biol. Med.* **2021**, *135*, 104558. [CrossRef]
35. Abdollahzadeh, B.; Gharehchopogh, F.S. A multi-objective optimization algorithm for feature selection problems. *Eng. Comput.* **2021**, 1–19. [CrossRef]
36. dos Santos Coelho, L.; Mariani, V.C. Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. *Expert Syst. Appl.* **2008**, *34*, 1905–1913. [CrossRef]
37. Zitzler, E.; Deb, K.; Thiele, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.* **2000**, *8*, 173–195. [CrossRef]
38. Zitzler, E.; Thiele, L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* **1999**, *3*, 257–271. [CrossRef]
39. Reyes-Sierra, M.; Coello CA, C. A study of fitness inheritance and approximation techniques for multi-objective particle swarm optimization. In Proceedings of the 2005 IEEE Congress on Evolutionary Computation, Edinburgh, UK, 2–5 September 2005; Volume 1, pp. 65–72.