mathematics

MDPI

Article

An Improved Wild Horse Optimizer for Solving Optimization Problems

Rong Zheng [1], Abdelazim G. Hussien [2,3,*], He-Ming Jia [1,*], Laith Abualigah [4,5], Shuang Wang [1] and Di Wu [6]

[1] School of Information Engineering, Sanming University, Sanming 365004, China; zhengr@fjsmu.edu.cn (R.Z.); wang.shuang@fjsmu.edu.cn (S.W.)
[2] Department of Computer and Information Science, Linköping University, 581 83 Linköping, Sweden
[3] Faculty of Science, Fayoum University, Faiyum 63514, Egypt
[4] Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan; aligah.2020@gmail.com
[5] School of Computer Science, Universiti Sains Malaysia, Gelugor 11800, Malaysia
[6] School of Education and Music, Sanming University, Sanming 365004, China; wudi@fjsmu.edu.cn
* Correspondence: abdelazim.hussien@liu.se (A.G.H.); jiaheming@fjsmu.edu.cn (H.-M.J.); Tel.: +86-4607-0425-0519 (A.G.H.); +86-132-0666-6920 (H.-M.J.)

Abstract: Wild horse optimizer (WHO) is a recently proposed metaheuristic algorithm that simulates the social behavior of wild horses in nature. Although WHO shows competitive performance compared to some algorithms, it suffers from low exploitation capability and stagnation in local optima. This paper presents an improved wild horse optimizer (IWHO), which incorporates three improvements to enhance optimizing capability. The main innovation of this paper is to put forward the random running strategy (RRS) and the competition for waterhole mechanism (CWHM). The random running strategy is employed to balance exploration and exploitation, and the competition for waterhole mechanism is proposed to boost exploitation behavior. Moreover, the dynamic inertia weight strategy (DIWS) is utilized to optimize the global solution. The proposed IWHO is evaluated using twenty-three classical benchmark functions, ten CEC 2021 test functions, and five real-world optimization problems. High-dimensional cases (D = 200, 500, 1000) are also tested. Comparing nine well-known algorithms, the experimental results of test functions demonstrate that the IWHO is very competitive in terms of convergence speed, precision, accuracy, and stability. Further, the practical capability of the proposed method is verified by the results of engineering design problems.

Keywords: wild horse optimizer; metaheuristic; optimization; exploration and exploitation; engineering design problem

MSC: 49

Citation: Zheng, R.; Hussien, A.G.; Jia, H.-M.; Abualigah, L.; Wang, S.; Wu, D. An Improved Wild Horse Optimizer for Solving Optimization Problems. *Mathematics* **2022**, *10*, 1311. https://doi.org/10.3390/math10081311

Academic Editors: Felix T.S. Chan, Erhan Kozan, Shi Qiang Liu and Weidong Li

Received: 30 March 2022
Accepted: 13 April 2022
Published: 14 April 2022

1. Introduction

Optimization is a research field that aims to maximize or minimize specific objective functions [1–3]. Most real-world engineering problems found in nature are classified as NP optimization problems [4,5]. Optimization exists in nearly every field, such as economics [6], text clustering [7], pattern recognition [8], chemistry [9], engineering design [10], feature selection [11,12], face detection and recognition [13], and information technology [14]. Classical and traditional mathematical approaches cannot solve these problems accurately or adequately. Recently, many researchers have tried to solve these problems using a new kind of approximation algorithm known as metaheuristics. In the last two decades, an enormous amount of development and work regarding metaheuristics algorithms have been done. Metaheuristics algorithms have received much attention and have become more popular due to their flexibility, simplicity, and their avoidance of local optima. Metaheuristics

algorithms simulate natural phenomena or physical law and can be generally categorized into four different classes: (1) swarm intelligence algorithms, (2) evolution algorithms, (3) physics and chemistry algorithms, and (4) human-based algorithms.

Swarm intelligence algorithms contain algorithms which simulate collective or social behavior of specific creatures. Creatures in the real world can interact with each other to achieve intelligent behavior. Examples of such algorithms include particle swarm optimization (PSO) [15], ant colony optimization (ACO) [16], cuckoo search (CS) [17], grasshopper optimization algorithm (GOA) [18], virus colony search (VCS) [19], dolphin echolocation (DE) algorithm [20], Harris Hawk optimization [21], whale optimization algorithm (WOA) [22], ant lion optimization (ALO) [23], crow search algorithm (CSA) [24], salp swarm algorithm (SSA) [25], moth–flame optimization (MFO) [26], snake optimizer (SO) [27], coot bird [28], and squirrel search algorithm (SSA) [29].

Evolution algorithms are population-based and stochastic algorithms which maintain biological evolution processes such as selection, mutation, elimination, and migration [12,13]. This class contains genetic algorithm (GA) [30], evolutionary programming (EP) [31], differential evolution (DE) [32], bacterial foraging optimization (BFO) [33], and memetic algorithm (MA) [34].

Physics and chemistry algorithms mimic chemical laws or physical phenomena in the universe. Examples of such algorithms are simulated annealing (SA) [35], gravitational search algorithm (GSA) [36], electromagnetism mechanism (EM) algorithm [37], ion motion algorithm (IMA) [38], lightning search algorithm (LSA) [39], and vortex search algorithm (VSA) [40].

Human-based algorithms: Humans are the most intelligent creature in the universe as they can easily find the best way to fix their problems and issues. Human-based algorithms contain algorithms that simulate physical or nonphysical activities such as thinking and physical activities. Examples of this class of algorithms include teaching–learning-based optimization (TLBO) [41], imperialist competitive algorithm (ICA) [42], and social-based algorithm (SBA) [43].

Although several metaheuristic techniques have been proposed, no free lunch (NFL) [44] encourages researchers to research more. NFL mentioned that no algorithm works efficiently in all optimization problems. If the algorithm can solve a specific type of problem efficiently, it may not solve other optimization problem classes. Therefore, many researchers devote themselves to proposing new algorithms or improving existing methods. For instance, Ning improved the whale optimization algorithm using Gaussian mutation [45]. Nautiyal and Tubishat enhanced the basic salp swarm algorithm by applying mutation schemes and opposition-based learning, respectively [46,47]. Moreover, Pelusi proposed a hybrid phase between exploration and exploitation using a fitness dependent weight factor to strengthen the balance between exploration and exploitation for moth–flame optimization [48]. Recently, a new optimization algorithm called wild horse optimizer (WHO) [49] was proposed by Naruei and Keynia. WHO simulates horse behavior in which they leave their group and join another group before becoming adults to prevent mating between siblings or daughters. However, WHO may fall in local optima regions or have a slow convergence in high-dimensional and complex problems.

In this paper, an improved version of WHO, called "IWHO", is proposed to enhance the original algorithm's effectiveness and performance. IWHO improved the original algorithm by using three different operators: random running strategy (RRS), dynamic inertia weight strategy (DIWS), and competition of waterhole mechanism (CWHM). In order to evaluate the performance of the WHO algorithm, we compared it with the classical WHO algorithm and eight other different algorithms: grey wolf optimizer (GWO) [50], moth–flame optimization (MFO) [51], salp swarm algorithm (SSA) [52], whale optimization algorithm (WOA) [53], particle swarm optimizer (PSO) [15], hybridizing sine–cosine algorithm with harmony search (HSCAHS) [54], dynamic sine–cosine algorithm (DSCA) [55], and modified ant lion optimizer (MALO) [56] using thirty-three benchmark functions. Moreover, five different constrained engineering problems were used: welded beam design,

tension/compression spring design, three-bar truss design, car crashworthiness design, and speed reducer design. The main contributions of this paper are:

- The random running strategy is proposed to balance the exploration and exploitation phases.
- The dynamic inertia weight strategy is applied to the waterhole to achieve the optimal global solution.
- The competition for waterhole mechanism is introduced to stallion position calculations to boost exploitation behavior.

This paper is organized as follows: Section 2 describes WHO and other used operators, whereas Section 3 describes the improved algorithm. Sections 4 and 5 show the experimental study and discussion using benchmark functions and five real-world engineering problems, whereas Section 6 concludes the paper.

2. Wild Horse Optimizer

Generally, horses can be divided into two classes based on their social organization (territorial and non-territorial). They live in groups of different ages, such as offspring, stallions, and mares (see Figure 1). Both stallions and mares live together and interact with each other in grazing. Foals leave their groups after they grow up and join other groups to establish their own families. This behavior prevents mating between stallions and siblings.

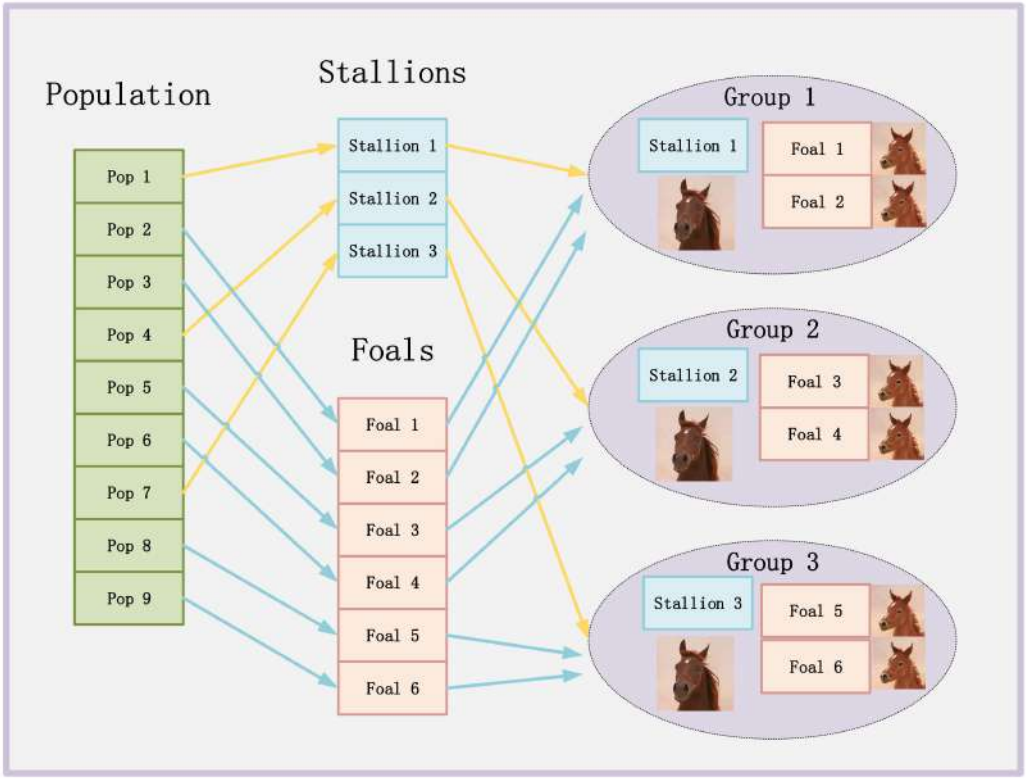

Figure 1. Formation of groups from the original population.

The wild horse optimization (WHO) algorithm is a metaheuristic swarm-based algorithm inspired by the social behavior of horses, such as grazing, domination, leadership hierarchy, and mating.

The WHO algorithm consists of five different steps, described below:

2.1. Creating Initial Populations, Horse Groups, Determining Leaders

If N individuals and G groups exist, then the number of non-leaders (mares and foals) is N–G, and the number of leaders is G. The proportion of stallions is defined as PS, which is G/N. Figure 1 shows how leaders are determined from the basic generation to create various groups.

2.2. Grazing Behavior

As stated previously, most of a foal's life is spent grazing near its group. In order to simulate the grazing phase, we assume that the stallion position existed in the grazing area center. The following formula is used to enable other individuals to move.

$$X_{G,j}^{i} = 2Z\cos(2\pi RZ) \times \left(\text{Stallion}_{G,j} - X_{G,j}^{i}\right) + \text{Stallion}_{G,j} \tag{1}$$

where $X_{G,j}^{i}$ and $\text{Stallion}_{G,j}$ are the positions of the ith group member and stallion in the jth group, respectively, R is a random number between -2 and 2, and Z is an adaptive parameter computed by Equation (2):

$$P = \vec{R1} < TDR,\ IDX = (P == 0),\ Z = R2\Theta IDX + \vec{R3}\Theta(\ \sim IDX) \tag{2}$$

where P is a vector containing 0 and 1, and its dimension equals the dimension of the problem, $\vec{R1}$ and $\vec{R3}$ are random vectors between 0 and 1, and R_2 is a random number between 0 and 1. TDR is a linearly decreasing parameter computed by Equation (3).

$$TDR = 1 - \frac{t}{T} \tag{3}$$

where t and T are the current and maximum iterations, respectively.

2.3. Horse Mating Behavior

As stated previously, one of the unique behaviors of horses compared to other animals is separating foals from their original groups prior to their reaching puberty and mating. To be able to simulate the behavior of mating between horses, the following formula is used:

$$\begin{gathered} X_{G,k}^{p} = \text{Crossover}\left(X_{G,i}^{q}, X_{G,j}^{z}\right), i \neq j \neq k, q = z = end \\ \text{Crossover} = \text{Mean} \end{gathered} \tag{4}$$

where $X_{G,k}^{p}$ is the position of horse p in group k, which is formed by positions of horse q in group i and horse z in group j. In the basic WHO, the probability of crossover is set to a constant named PC.

2.4. Group Leadership

Group leaders (stallions) will lead other group members to a suitable area (waterhole). Group leaders (stallions) will also compete for the waterhole, leading the dominant group to employ the waterhole first. The following formula is used to simulate this behavior:

$$\overline{\text{Stallion}}_{G,j} = \begin{cases} 2Z\cos(2\pi RZ) \times \left(\text{WH} - \text{Stallion}_{G,j}\right) + \text{WH}\ if\ rand > 0.5 \\ 2Z\cos(2\pi RZ) \times \left(\text{WH} - \text{Stallion}_{G,j}\right) - \text{WH}\ if\ rand \leq 0.5 \end{cases} \tag{5}$$

where $\overline{\text{Stallion}}_{G,j}$ and $\text{Stallion}_{G,j}$ are the candidate position and the current leader position in the jth group, respectively, and WH is the position of the waterhole.

2.5. Exchange and Selection of Leaders

At first, leaders are selected randomly. After that, leaders are selected based on their fitness values. To simulate the exchange between leader positions and other individuals, the following formula is used:

$$\text{Stallion}_{G,j} = \begin{cases} X^i_{G,j}, if\ f\left(X^i_{G,j}\right) < f\left(\text{Stallion}_{G,j}\right) \\ \text{Stallion}_{G,i}, if\ f\left(X^i_{G,j}\right) \geq f\left(\text{Stallion}_{G,j}\right) \end{cases} \tag{6}$$

where $f(X^i_{G,j})$ and $f(\text{Stallion}_{G,j})$ are the fitness values of foal and stallion, respectively.

3. Improved Wild Horse Optimizer

In this section, the IWHO is introduced in detail. To enhance the optimizing capability of the basic algorithm, three methods are added. Firstly, wild horses in nature tend to chase and run. Thus, a random running strategy is applied to the foals and stallions. Then, a dynamic inertia weight strategy is introduced to the waterhole, which is from DSCA [50] and beneficial to balance exploration and exploitation. At last, the competition for waterhole mechanism is proposed for stallions, which is inspired by artificial gorilla troops optimizer (GTO) [57] and further improves the solution quality.

3.1. Random Running Strategy (RRS)

In nature, wild horses are very fond of running or chasing each other. Inspired by this, the random running strategy is proposed for both foals and stallions. The position-updating formula is presented as follow:

$$X^i_{G,j} \text{ or } \overline{\text{Stallion}}_{G,j} = lb + (ub - lb) \times rand \tag{7}$$

where lb and ub are the lower boundary and upper boundary, respectively.

When conducting the RRS, search agents may appear anywhere in the search space. Thus, this method can help search agents jump out of the local optima. Note that to balance exploration and exploitation, the probability of random running (*PRR*) is set to a small value of 0.1.

3.2. Dynamic Inertia Weight Strategy (DIWS)

The dynamic weight strategy has been applied in much research [55,58,59] and is helpful to find the optimal global solution when introduced into algorithms. Thus, to help the stallions find a better waterhole, a dynamic inertia weight is added to the waterhole in the first formula of Equation (5). The weight and modified formula are calculated as follows:

$$w = \begin{cases} w\ \min + (w\ \max - w\ \min) \times \frac{f(t)i - f(t)\min}{f(t)avg - f(t)\min}, \ if\ f(t)i \leq f(t)avg \\ w\ \max, \ if\ f(t)i > f(t)avg \end{cases} \tag{8}$$

$$\overline{\text{Stallion}}_{G,j} = 2Z\cos(2\pi RZ) \times \left(\text{WH} - \text{Stallion}_{G,j}\right) + w \times \text{WH} \tag{9}$$

where $w_{\min}$ and $w_{\max}$ are the upper and lower boundary values, respectively, $f(t)_i$ is the fitness value of the current stallion at tth iteration, $f(t)_{avg}$ is the average fitness value of all stallions, and $f(t)_{min}$ is the minimum fitness value of the population.

3.3. Competition for Waterhole Mechanism (CWHM)

Moreover, the waterhole can be found by two stallions simultaneously. Then there may be rivalry between stallions over the waterhole. Thus, to further improve the solution quality, the competition for the waterhole mechanism, which is similar to the competition

for adult females in GTO [53], is proposed to replace the second formula of Equation (5). The position updating formula is as follows:

$$\overline{\text{Stallion}}_{G,j} = \text{WH} - Z \times \left(\text{Stallion}_{G,j} \times Q_1 - \text{Stallion}_{G,j} \times Q_2\right) \tag{10}$$

where Q_1 and Q_2 are random numbers between -1 and 1.

3.4. Improved Wild Horse Optimizer

By combining the DIWS and CWHM, the following formula is used to replace Equation (5) in WHO:

$$\overline{\text{Stallion}}_{G,j} = \begin{cases} 2Z\cos(2\pi RZ) \times \left(\text{WH} - \text{Stallion}_{G,j}\right) + w \times \text{WH}, if\ R_3 > 0.5 \\ \text{WH} - Z \times \left(\text{Stallion}_{Gi} \times Q_1 - \text{Stallion}_{Gj} \times Q_2\right), if\ R_3 \leq 0.5 \end{cases} \tag{11}$$

In IWHO, foals and stallions have a more flexible method to update their positions. The RRS helps search agents achieve a better balance between exploration and exploitation. At the same time, DIWS and CWHM enable the method to obtain high-quality solutions and improve the convergence speed. The pseudocode of IWHO is shown in Algorithm 1, and the flowchart of the proposed IWHO is shown in Figure 2.

Algorithm 1 Pseudocode of IWHO

Start IWHO.
Input IWHO parameters: PC = 0.13, PS = 0.2, $w_{\min}$ = 0.01, $w_{\max}$ = 0.99, PRR = 0.1.
Set population size (N) and the maximum number of iterations (T).
Initialize the population of horses at random.
Create foal groups and select stallions.
While ($t \leq T$)
  Calculate TDR using Equation (3).
  Calculate Z using Equation (2).
  For the number of stallions
   For the number of foals
    If *rand* > PC
     Update the position of the foal using Equation (1).
    Else if *rand* > PRR
     Update the position of the foal using Equation (4).
    Else
     Update the position of the foal using Equation (7).
    End
   End for
   If *rand* > PRR
    If *rand* > 0.5
     Generate the candidate position of stallion using Equation (10).
    Else
     Generate the candidate position of stallion using Equation (9).
    End if
   Else
    Generate the candidate position of stallion using Equation (7).
   End if
   If the candidate position of the stallion is better
    Replace the position of the stallion using the candidate position.
   End if
  End for
  Exchange foals and stallions position using Equation (6).
  $t = t + 1$
End While
Output the best solution obtained by IWHO.
End IWHO.

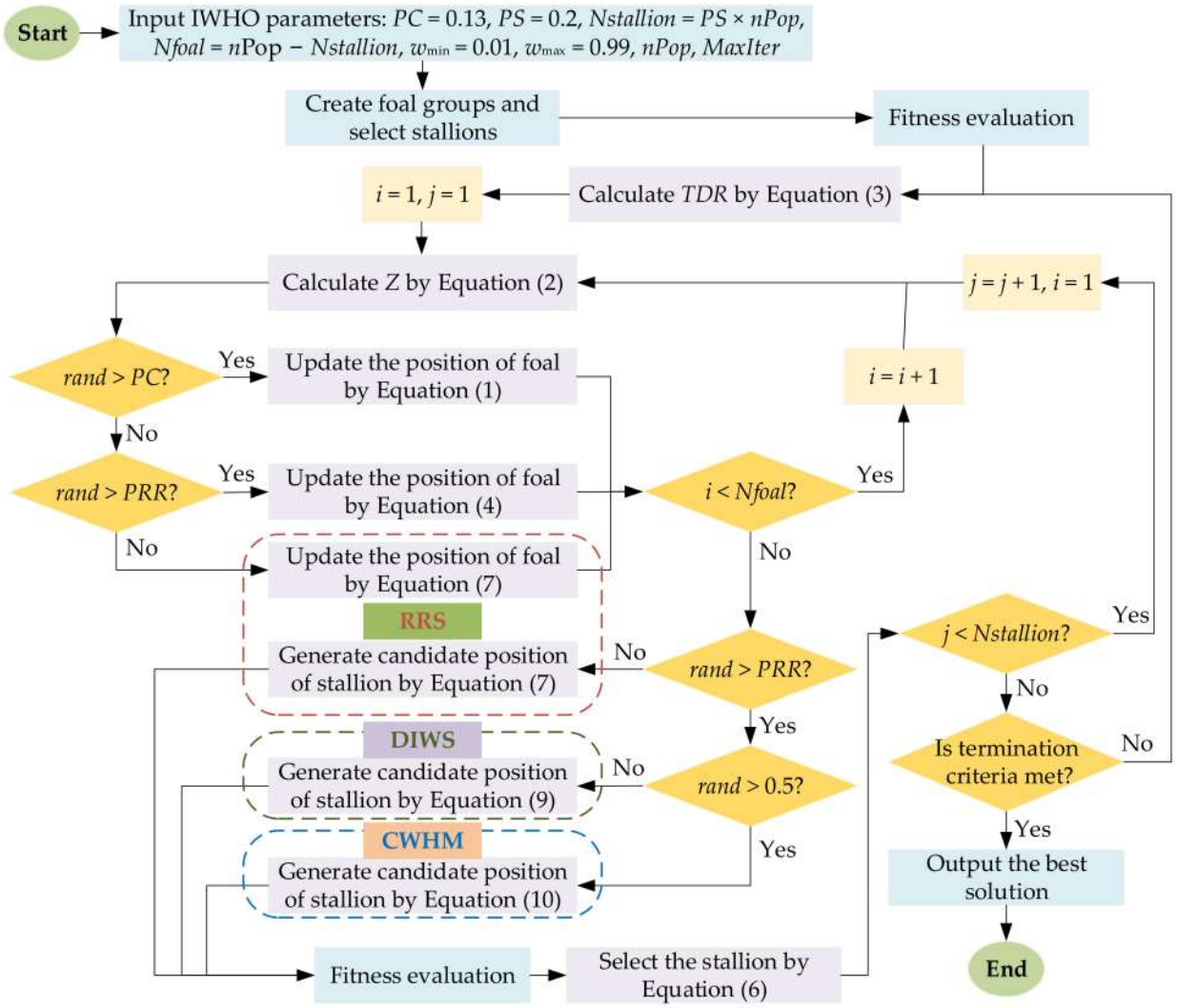

Figure 2. Flowchart of IWHO.

3.5. Analysis of Algorithm Computational Complexity

The computational complexity of IWHO is related to the population size (N), the maximum number of iterations (T), and problem dimensions (D). In the basic WHO, the computational complexity of initializing population is $O(N \times D)$. Then, the computational complexity of updating positions for foals and stallions is $O(N \times D \times T)$. In addition, consider the worst-case scenario: all foals conduct the grazing behavior. The computational complexity of calculating Z is $O(N \times D \times T)$. The computational complexity of the exchange and selection of leaders is $O(Nfoal \times T)$. Thus, the overall computational complexity of WHO is $O(2 \times N \times D \times T + N \times D + Nfoal \times T)$. In IWHO, it should be noted that the proposed RRS and CWHM would not increase the computational complexity.

Similarly, considering the worst-case scenario, all stallions conduct the DIWS. Then, the computational complexity of calculating weight is $O(Nstallion \times T)$. Therefore, the overall computational complexity of IWHO is $O(2 \times N \times D \times T + N \times D + N \times T)$, which is only a little higher than the basic algorithm.

4. Experimental Study

To be able to show the significant performance and superiority of IWHO, a variety of experiments were carried out. IWHO was tested over 33 mathematical functions taken from two well-known benchmark function datasets: CEC2005 [60] and CEC2021 [61]. IWHO performance was compared to classical WHO and eight different algorithms. The parameter settings of these experiments are discussed in the following subsection.

4.1. Optimization of Functions and Parameter Settings

Thirty-three mathematical functions of different types (unimodal, multimodal, fixed-dimension multimodal functions) were implemented from two different benchmark datasets: CEC2005 and CEC2021. Table 1 shows each mathematical function used to carry out these experiments, including its type, range of the search space, and its global optimal value.

Table 1. Feature properties of the test functions (*D* indicates the dimension).

Function Type	Function	*D*	Range	Theoretical Optimization Value
Unimodal test functions	F1	30/100/500/1000	[−100, 100]	0
	F2	30/100/500/1000	[−10, 10]	0
	F3	30/100/500/1000	[−100, 100]	0
	F4	30/100/500/1000	[−100, 100]	0
	F5	30/100/500/1000	[−30, 30]	0
	F6	30/100/500/1000	[−100, 100]	0
	F7	30/100/500/1000	[−1.28, 1.28]	0
Multimodal test functions	F8	30/100/500/1000	[−500, 500]	$-418.9829 \times D$
	F9	30/100/500/1000	[−5.12, 5.12]	0
	F10	30/100/500/1000	[−32, 32]	0
	F11	30/100/500/1000	[−600, 600]	0
	F12	30/100/500/1000	[−50, 50]	0
	F13	30/100/500/1000	[−50, 50]	0
Fixed-dimension multimodal test functions	F14	2	[−65, 65]	0.998004
	F15	4	[−5, 5]	0.0003075
	F16	2	[−5, 5]	−1.03163
	F17	2	[−5, 5]	0.398
	F18	2	[−2, 2]	3
	F19	3	[−1, 2]	−3.8628
	F20	6	[0, 1]	−3.3220
	F21	4	[0, 10]	−10.1532
	F22	4	[0, 10]	−10.4028
	F23	4	[0, 10]	−10.5363
CEC2021 unimodal test functions	CEC_01	20	[−100, 100]	100
CEC2021 basic test functions	CEC_02	20	[−100, 100]	1100
	CEC_03	20	[−100, 100]	700
	CEC_04	20	[−100, 100]	1900
CEC2021 hybrid test functions	CEC_05	20	[−100, 100]	1700
	CEC_06	20	[−100, 100]	1600
	CEC_07	20	[−100, 100]	2100
CEC2021 composition test functions	CEC_08	20	[−100, 100]	2200
	CEC_09	20	[−100, 100]	2400
	CEC_10	20	[−100, 100]	2500

IWHO has been compared to WHO and eight other algorithms: grey wolf optimizer (GWO) [50], moth–flame optimization (MFO) [51], salp swarm algorithm (SSA) [52], whale optimization algorithm (WOA) [53], particle swarm optimizer (PSO) [15], hybrid sine–cosine algorithm with harmony search (HSCAHS) [54], dynamic sine–cosine algorithm (DSCA) [55], and modified ant lion optimizer (MALO) [56]. The parameter settings of each algorithm are shown in Table 2. These comparative algorithms have shown certain optimizing capability to the benchmark functions. However, for some optimization problems, they cannot get satisfactory optimization solutions. Hence, the advantages of the improved algorithm can be revealed through the extensive comparison between the improved algorithm and other algorithms.

Table 2. Parameter values for the IWHO and other comparative optimization algorithms.

Algorithm	Parameters
IWHO	$PS = 0.2$; $PC = 0.13$; $PRR = 0.1$; $w \in [0.01, 0.99]$
WHO [49]	$PS = 0.2$; $PC = 0.13$
GWO [50]	$a = [2, 0]$
MFO [51]	$b = 1$; $t = [-1,1]$; $a \in [-1,-2]$
SSA [52]	$c_1 \in [0, 1]$; $c_2 \in [0, 1]$
WOA [53]	$a_1 = [2, 0]$; $a_2 = [-2, -1]$; $b = 1$
PSO [15]	$c_1 = 2$; $c_2 = 2$; $W \in [0.2, 0.9]$; $vMax = 6$
HSCAHS [54]	$a = 2$; $Bandwidth = 0.02$
DSCA [55]	$w \in [0.1, 0.9]$, $\sigma = 0.1$; $a_{end} = 0$; $a_{start} = 2$
MALO [56]	$Switch\ possibility = 0.5$

The experiments were carried out in Windows 10 with 16 GB RAM and an Intel (R) i5-9500 CPU. All simulations were executed using MATLAB R2016a. Moreover, the maximum number of iterations was set at 500, and the number of individuals was set to 30. In contrast, the number of dimensions was different according to each function, as shown in Table 1.

4.2. Analysis of Performance for the CEC 2005 Test Suite

Improved WHO (IWHO) results with other comparative algorithms are listed in Table 3. This table shows that the suggested approach achieves the best results in almost all unimodal functions (F1–F7) in all mentioned dimensions (30, 200, 500, and 1000). On the other hand, IWHO also ranked first in almost multimodal functions (F8–F13), and in almost functions regarding fixed-dimension multimodal functions (F14–F23). Compared to basic WHO and other algorithms, it can be seen that the improved algorithm has better exploration and exploitation capability as a result of the three improvements.

Moreover, a non-parametric test known as Wilcoxon signed test was used to give evidence of the performance of our suggested approach. Table 4 shows the Wilcoxon results of IWHO vs. other algorithms with 5% significance [62]. From this table, we can see that IWHO is better than many other metaheuristics algorithms in most cases.

Furthermore, Figure 3 shows the convergence curves of some functions (F1–F3, F5, F6, F8, F10, F12, F13) with 30 dimensions. It is clear from this figure that IWHO has powerful convergence. Further, Figures 4–6 show the convergence curves of the same functions (F1–F3, F5, F6, F8, F10, F12, F13) with dimensions of 200, 500, and 1000, respectively. Figure 7 shows the convergence curves for some fixed-dimension multimodal functions (F14, F15, F17–F23). It is obvious that the suggested algorithm has good convergence even in these multimodal functions.

Table 3. Comparison of optimization results between IWHO and other optimization algorithms on classical test functions (F1−F23).

Function	*D*	Metric	IWHO	WHO	GWO	MFO	SSA	WOA	PSO	HSCAHS	DSCA	MALO
F1	30	Mean	0	3.05×10^{-43}	2.09×10^{-27}	1.69×10^{3}	1.97×10^{-7}	8.09×10^{-74}	2.09×10^{-4}	2.79×10^{-50}	3.88×10^{-106}	1.28×10^{-3}
		Std	0	1.57×10^{-42}	3.99×10^{-27}	3.83×10^{3}	3.38×10^{-7}	2.79×10^{-73}	3.23×10^{-4}	1.41×10^{-49}	2.11×10^{-105}	1.17×10^{-3}
	200	Mean	0	4.57×10^{-32}	9.50×10^{-8}	2.88×10^{5}	1.69×10^{4}	5.20×10^{-70}	3.38×10^{2}	3.46×10^{-34}	1.34×10^{-91}	3.95×10^{4}
		Std	0	1.81×10^{-31}	4.45×10^{-8}	2.81×10^{4}	2.15×10^{3}	2.84×10^{-69}	4.88×10^{1}	8.54×10^{-34}	7.35×10^{-91}	9.15×10^{3}
	500	Mean	0	2.96×10^{-28}	1.63×10^{-3}	1.16×10^{6}	9.46×10^{4}	3.14×10^{-71}	5.85×10^{3}	4.56×10^{-29}	1.90×10^{-78}	2.17×10^{5}
		Std	0	1.62×10^{-27}	5.53×10^{-4}	3.51×10^{4}	6.80×10^{3}	8.49×10^{-71}	4.89×10^{2}	8.10×10^{-29}	1.04×10^{-77}	3.75×10^{4}
	1000	Mean	0	6.50×10^{-29}	2.53×10^{-1}	2.73×10^{6}	2.37×10^{5}	6.48×10^{-70}	4.10×10^{4}	2.05×10^{-25}	4.69×10^{-73}	5.80×10^{5}
		Std	0	2.29×10^{-28}	6.74×10^{-2}	5.33×10^{4}	1.27×10^{4}	3.21×10^{-69}	2.39×10^{3}	4.20×10^{-25}	2.57×10^{-72}	8.66×10^{4}
F2	30	Mean	0	2.22×10^{-25}	9.06×10^{-17}	2.65×10^{1}	2.37	1.56×10^{-51}	7.36	4.04×10^{-27}	1.22×10^{-60}	5.29×10^{1}
		Std	0	7.39×10^{-25}	6.90×10^{-17}	2.03×10^{1}	1.73	4.94×10^{-51}	1.05×10^{1}	9.46×10^{-27}	6.65×10^{-60}	5.31×10^{1}
	200	Mean	0	6.92×10^{1}	3.14×10^{-5}	7.55×10^{2}	1.55×10^{2}	1.18×10^{-49}	4.76×10^{2}	1.19×10^{-18}	1.05×10^{-51}	3.76×10^{73}
		Std	0	1.80×10^{2}	9.14×10^{-6}	4.70×10^{1}	1.26×10^{1}	3.91×10^{-49}	6.08×10^{1}	1.99×10^{-18}	3.96×10^{-51}	2.06×10^{74}
	500	Mean	0	6.01×10^{2}	1.09×10^{-2}	4.30×10^{116}	5.38×10^{2}	1.00×10^{-48}	5.74×10^{58}	3.84×10^{-16}	7.18×10^{-50}	2.31×10^{244}
		Std	0	6.18×10^{2}	2.39×10^{-3}	2.23×10^{117}	2.39×10^{1}	4.88×10^{-48}	3.14×10^{59}	6.21×10^{-16}	3.93×10^{-49}	Inf
	1000	Mean	0	1.43×10^{3}	8.06×10^{-1}	Inf	1.20×10^{3}	1.30×10^{-49}	1.40×10^{3}	5.11×10^{-15}	Inf	Inf
		Std	0	1.37×10^{3}	9.03×10^{-1}	NaN	3.61×10^{1}	3.29×10^{-49}	7.08×10^{1}	1.09×10^{-14}	NaN	NaN
F3	30	Mean	0	3.56×10^{-26}	6.23×10^{-6}	2.28×10^{4}	1.74×10^{3}	4.35×10^{4}	7.36×10^{1}	1.08×10^{-48}	2.25×10^{-57}	4.56×10^{3}
		Std	0	1.82×10^{-25}	1.02×10^{-5}	1.31×10^{4}	1.30×10^{3}	1.17×10^{4}	3.10×10^{1}	4.06×10^{-48}	1.23×10^{-56}	1.79×10^{3}
	200	Mean	0	1.76×10^{-12}	2.14×10^{4}	8.63×10^{5}	2.32×10^{5}	5.05×10^{6}	8.90×10^{4}	1.73×10^{-31}	2.86×10^{-45}	3.42×10^{5}
		Std	0	4.91×10^{-12}	1.11×10^{4}	1.56×10^{5}	1.31×10^{5}	1.36×10^{6}	1.88×10^{4}	5.48×10^{-31}	1.54×10^{-44}	9.60×10^{4}
	500	Mean	0	1.06×10^{-8}	3.26×10^{5}	4.86×10^{6}	1.01×10^{6}	2.91×10^{7}	5.94×10^{5}	5.34×10^{-26}	1.35×10^{-43}	2.09×10^{6}
		Std	0	5.30×10^{-8}	1.03×10^{5}	9.34×10^{5}	5.23×10^{5}	9.83×10^{6}	1.18×10^{5}	9.19×10^{-26}	6.83×10^{-43}	6.22×10^{5}
	1000	Mean	0	2.36×10^{-5}	1.52×10^{6}	1.83×10^{7}	4.24×10^{6}	1.18×10^{8}	2.47×10^{6}	5.52×10^{-23}	1.46×10^{-35}	7.32×10^{6}
		Std	0	1.28×10^{-4}	2.93×10^{5}	4.06×10^{6}	1.96×10^{6}	4.57×10^{7}	6.38×10^{5}	1.12×10^{-22}	7.85×10^{-35}	2.30×10^{6}
F4	30	Mean	1.98×10^{-320}	1.97×10^{-17}	7.10×10^{-7}	6.92×10^{1}	1.05×10^{1}	5.04×10^{1}	1.04	9.05×10^{-26}	5.55×10^{-48}	1.68×10^{1}
		Std	0	4.78×10^{-17}	6.47×10^{-7}	9.08	2.71	2.89×10^{1}	2.44×10^{-1}	2.18×10^{-25}	3.03×10^{-47}	3.47
	200	Mean	0	2.10×10^{-11}	2.37×10^{1}	9.72×10^{1}	3.49×10^{1}	8.49×10^{1}	1.96×10^{1}	2.86×10^{-16}	1.39×10^{-41}	4.09×10^{1}
		Std	0	7.03×10^{-11}	6.67	7.39×10^{-1}	3.49	1.40×10^{1}	1.61	4.71×10^{-16}	5.77×10^{-41}	4.46
	500	Mean	0	7.02×10^{-10}	6.47×10^{1}	9.89×10^{1}	4.00×10^{1}	7.72×10^{1}	2.81×10^{1}	1.16×10^{-11}	3.39×10^{-33}	4.95×10^{1}
		Std	0	1.24×10^{-9}	4.89	4.11×10^{-1}	3.33	2.80×10^{1}	1.37	1.59×10^{-11}	1.73×10^{-32}	4.89
	1000	Mean	0	2.25×10^{-9}	7.89×10^{1}	9.95×10^{1}	4.55×10^{1}	8.43×10^{1}	3.30×10^{1}	2.50×10^{-7}	1.97×10^{-28}	5.30×10^{1}
		Std	0	4.00×10^{-9}	3.18	1.95×10^{-1}	3.35	1.85×10^{1}	1.90	4.41×10^{-7}	1.08×10^{-27}	4.17

Table 3. *Cont.*

Function	*D*	Metric	IWHO	WHO	GWO	MFO	SSA	WOA	PSO	HSCAHS	DSCA	MALO
F5	30	Mean	4.37	3.58×10^{1}	2.70×10^{1}	2.68×10^{6}	2.19×10^{2}	2.79×10^{1}	3.13×10^{3}	2.88×10^{1}	2.86×10^{1}	9.77×10^{-1}
		Std	9.71	2.35×10^{1}	6.78×10^{-1}	1.46×10^{7}	3.37×10^{2}	4.26×10^{-1}	1.64×10^{4}	9.01×10^{-2}	3.04×10^{-1}	5.22
	200	Mean	4.34×10^{1}	2.88×10^{2}	1.98×10^{2}	1.05×10^{9}	4.11×10^{6}	1.98×10^{2}	6.61×10^{5}	1.99×10^{2}	1.99×10^{2}	5.60×10^{1}
		Std	7.82×10^{1}	3.26×10^{2}	4.76×10^{-1}	1.13×10^{8}	1.14×10^{6}	1.89×10^{-1}	1.52×10^{5}	4.40×10^{-2}	6.75×10^{-2}	4.75×10^{1}
	500	Mean	1.27×10^{2}	4.99×10^{2}	4.98×10^{2}	5.12×10^{9}	3.86×10^{7}	4.96×10^{2}	3.00×10^{7}	4.99×10^{2}	4.99×10^{2}	2.31×10^{2}
		Std	1.90×10^{2}	2.18	3.18×10^{-1}	2.17×10^{8}	8.07×10^{6}	4.62×10^{-1}	3.48×10^{6}	5.15×10^{-3}	1.87×10^{-2}	1.93×10^{2}
	1000	Mean	1.22×10^{2}	9.98×10^{2}	1.07×10^{3}	1.24×10^{10}	1.18×10^{8}	9.94×10^{2}	2.82×10^{8}	9.99×10^{2}	9.99×10^{2}	4.36×10^{2}
		Std	2.47×10^{2}	9.67×10^{-1}	3.20×10^{1}	3.12×10^{8}	1.13×10^{7}	7.27×10^{-1}	2.87×10^{7}	3.67×10^{-2}	1.69×10^{-2}	3.94×10^{2}
F6	30	Mean	5.09×10^{-5}	9.53×10^{-3}	8.12×10^{-1}	1.00×10^{3}	1.52×10^{-7}	4.21×10^{-1}	2.27×10^{-4}	6.67	5.57	4.69×10^{-4}
		Std	4.64×10^{-5}	2.92×10^{-2}	4.15×10^{-1}	3.04×10^{3}	1.80×10^{-7}	2.56×10^{-1}	3.22×10^{-4}	1.76×10^{-1}	3.09×10^{-1}	3.21×10^{-4}
	200	Mean	2.19	3.58×10^{1}	2.91×10^{1}	2.88×10^{5}	1.69×10^{4}	1.07×10^{1}	3.24×10^{2}	4.92×10^{1}	4.81×10^{1}	1.56
		Std	3.26	3.22×10^{1}	1.58	2.26×10^{4}	1.84×10^{3}	2.78	4.57×10^{1}	2.13×10^{-1}	3.38×10^{-1}	6.98×10^{-1}
	500	Mean	1.07×10^{1}	1.10×10^{2}	9.06×10^{1}	1.18×10^{6}	9.48×10^{4}	2.84×10^{1}	5.92×10^{3}	1.24×10^{2}	1.23×10^{2}	6.01
		Std	1.34×10^{1}	3.18×10^{1}	1.89	2.78×10^{4}	5.33×10^{3}	8.78	5.16×10^{2}	2.13×10^{-1}	3.46×10^{-1}	2.62
	1000	Mean	1.20×10^{1}	2.29×10^{2}	2.02×10^{2}	2.74×10^{6}	2.36×10^{5}	6.85×10^{1}	4.06×10^{4}	2.49×10^{2}	2.48×10^{2}	1.24×10^{1}
		Std	2.28×10^{1}	2.01×10^{1}	2.53	3.49×10^{4}	1.03×10^{4}	1.80×10^{1}	1.94×10^{3}	2.60×10^{-1}	4.41×10^{-1}	4.84
F7	30	Mean	2.23×10^{-4}	1.23×10^{-3}	1.85×10^{-3}	2.29	1.77×10^{-1}	4.07×10^{-3}	3.16	1.30×10^{-4}	1.86×10^{-3}	9.01×10^{-5}
		Std	2.05×10^{-4}	8.00×10^{-4}	1.19×10^{-3}	2.96	7.74×10^{-2}	5.54×10^{-3}	4.20	1.92×10^{-4}	1.88×10^{-3}	7.28×10^{-5}
	200	Mean	4.58×10^{-4}	1.62×10^{-3}	1.70×10^{-2}	2.99×10^{3}	1.76×10^{1}	6.24×10^{-3}	2.92×10^{3}	1.08×10^{-4}	2.34×10^{-3}	7.14×10^{-5}
		Std	4.58×10^{-4}	2.22×10^{-3}	5.45×10^{-3}	4.19×10^{2}	3.53	7.00×10^{-3}	4.63×10^{2}	9.81×10^{-5}	2.07×10^{-3}	5.22×10^{-5}
	500	Mean	5.06×10^{-4}	1.75×10^{-3}	5.38×10^{-2}	3.86×10^{4}	2.72×10^{2}	3.39×10^{-3}	4.39×10^{4}	1.13×10^{-4}	3.77×10^{-3}	1.03×10^{-4}
		Std	6.25×10^{-4}	1.56×10^{-3}	1.48×10^{-2}	1.74×10^{3}	4.13×10^{1}	3.30×10^{-3}	5.70×10^{3}	8.65×10^{-5}	3.35×10^{-3}	1.10×10^{-4}
	1000	Mean	4.28×10^{-4}	3.08×10^{-3}	1.38×10^{-1}	1.98×10^{5}	1.78×10^{3}	5.79×10^{-3}	2.44×10^{5}	1.14×10^{-4}	3.20×10^{-3}	9.34×10^{-5}
		Std	4.57×10^{-4}	2.65×10^{-3}	2.96×10^{-2}	7.27×10^{3}	1.74×10^{2}	5.81×10^{-3}	6.04×10^{3}	1.43×10^{-4}	3.32×10^{-3}	8.66×10^{-5}
F8	30	Mean	-1.26×10^{4}	-8.80×10^{3}	-5.98×10^{3}	-8.46×10^{3}	-7.46×10^{3}	-9.80×10^{3}	-4.77×10^{3}	-2.44×10^{3}	-4.46×10^{3}	-1.21×10^{4}
		Std	6.64×10^{-4}	5.09×10^{2}	6.98×10^{2}	1.19×10^{3}	7.32×10^{2}	1.57×10^{3}	1.30×10^{3}	3.14×10^{2}	3.59×10^{2}	1.08×10^{3}
	200	Mean	-8.38×10^{4}	-3.48×10^{4}	-2.96×10^{4}	-3.70×10^{4}	-3.46×10^{4}	-6.87×10^{4}	-1.31×10^{4}	-6.34×10^{3}	-1.18×10^{4}	-7.52×10^{4}
		Std	5.24	3.54×10^{3}	4.82×10^{3}	2.68×10^{3}	3.14×10^{3}	1.29×10^{4}	4.80×10^{3}	9.93×10^{2}	8.64×10^{2}	9.10×10^{3}
	500	Mean	-2.09×10^{5}	-6.03×10^{4}	-5.72×10^{4}	-6.21×10^{4}	-5.99×10^{4}	-1.77×10^{5}	-2.40×10^{4}	-1.05×10^{4}	-1.91×10^{4}	-1.90×10^{5}
		Std	3.87×101	7.66×10^{3}	4.22×10^{3}	4.14×10^{3}	4.63×10^{3}	2.92×10^{4}	8.68×10^{3}	1.56×10^{3}	2.27×10^{3}	2.43×10^{4}
	1000	Mean	-4.19×10^{5}	-8.36×10^{4}	-8.80×10^{4}	-8.87×10^{4}	-8.66×10^{4}	-3.48×10^{5}	-3.44×10^{4}	-1.52×10^{4}	-2.83×10^{4}	-3.76×10^{5}
		Std	2.01×10^{2}	9.06×10^{3}	1.43×10^{4}	6.25×10^{3}	6.65×10^{3}	5.54×10^{4}	1.30×10^{4}	2.13×10^{3}	3.81×10^{3}	4.60×10^{4}

Table 3. *Cont.*

Function	*D*	Metric	IWHO	WHO	GWO	MFO	SSA	WOA	PSO	HSCAHS	DSCA	MALO
F9	30	Mean	0	4.55×10^{-10}	3.40	1.66×10^{2}	5.25×10^{1}	0	1.01×10^{2}	0	0	8.33×10^{1}
		Std	0	1.74×10^{-9}	4.19	3.80×10^{1}	1.57×10^{1}	0	3.03×10^{1}	0	0	2.80×10^{1}
	200	Mean	0	0	2.21×10^{1}	2.22×10^{3}	8.20×10^{2}	7.58×10^{-15}	2.03×10^{3}	0	0	1.03×10^{3}
		Std	0	0	9.22	1.11×10^{2}	8.14×10^{1}	4.15×10^{-14}	1.45×10^{2}	0	0	8.92×10^{1}
	500	Mean	0	0	7.71×10^{1}	6.98×10^{3}	3.15×10^{3}	0	6.31×10^{3}	0	0	3.74×10^{3}
		Std	0	0	2.84×10^{1}	1.37×10^{2}	1.38×10^{2}	0	1.92×10^{2}	0	0	1.88×10^{2}
	1000	Mean	0	0	2.05×10^{2}	1.55×10^{4}	7.63×10^{3}	0	1.42×10^{4}	0	0	8.78×10^{3}
		Std	0	0	4.77×10^{1}	2.12×10^{2}	1.44×10^{2}	0	3.30×10^{2}	0	0	2.93×10^{2}
F10	30	Mean	8.88×10^{-16}	2.07×10^{-15}	1.03×10^{-13}	1.78×10^{1}	2.74	4.44×10^{-15}	4.07×10^{-1}	8.88×10^{-16}	8.88×10^{-16}	4.17
		Std	0	1.70×10^{-15}	2.27×10^{-14}	4.32	7.62×10^{-1}	1.87×10^{-15}	6.82×10^{-1}	0	0	2.11
	200	Mean	8.88×10^{-16}	2.66×10^{-15}	2.40×10^{-5}	2.00×10^{1}	1.31×10^{1}	4.44×10^{-15}	6.57	8.88×10^{-16}	8.88×10^{-16}	1.49×10^{1}
		Std	0	5.26×10^{-15}	4.81×10^{-6}	7.54×10^{-2}	6.63×10^{-1}	2.47×10^{-15}	3.42×10^{-1}	0	0	8.88×10^{-1}
	500	Mean	8.88×10^{-16}	1.48×10^{-15}	1.89×10^{-3}	2.03×10^{1}	1.42×10^{1}	5.15×10^{-15}	1.20×10^{1}	1.24×10^{-15}	8.88×10^{-16}	1.60×10^{1}
		Std	0	1.35×10^{-15}	4.06×10^{-4}	1.17×10^{-1}	2.52×10^{-1}	2.86×10^{-15}	3.13×10^{-1}	1.08×10^{-15}	0	7.80×10^{-1}
	1000	Mean	8.88×10^{-16}	1.95×10^{-15}	1.76×10^{-2}	2.03×10^{1}	1.46×10^{1}	4.56×10^{-15}	1.60×10^{1}	6.10×10^{-15}	8.88×10^{-16}	1.60×10^{1}
		Std	0	1.90×10^{-15}	2.26×10^{-3}	2.24×10^{-1}	1.45×10^{-1}	2.55×10^{-15}	3.82×10^{-1}	8.64×10^{-15}	0	1.28
F11	30	Mean	0	0	2.56×10^{-3}	3.10×10^{1}	2.07×10^{-2}	8.30×10^{-3}	9.87×10^{-3}	0	0	5.55×10^{-2}
		Std	0	0	5.97×10^{-3}	4.92×10^{1}	1.48×10^{-2}	3.16×10^{-2}	1.05×10^{-2}	0	0	2.88×10^{-2}
	200	Mean	0	0	5.28×10^{-3}	2.56×10^{3}	1.57×10^{2}	0	1.48	0	0	3.74×10^{2}
		Std	0	0	1.37×10^{-2}	1.65×10^{2}	2.56×10^{1}	0	6.05×10^{-1}	0	0	9.11×10^{1}
	500	Mean	0	0	2.22×10^{-2}	1.03×10^{4}	8.55×10^{2}	3.70×10^{-18}	8.09×10^{1}	0	0	1.96×10^{3}
		Std	0	0	3.95×10^{-2}	3.03×10^{2}	5.06×10^{1}	2.03×10^{-17}	9.46	0	0	3.25×10^{2}
	1000	Mean	0	0	3.75×10^{-2}	2.45×10^{4}	2.13×10^{3}	3.70×10^{-18}	2.77×10^{2}	0	0	5.35×10^{3}
		Std	0	0	5.93×10^{-2}	4.32×10^{2}	1.03×10^{2}	2.03×10^{-17}	2.08×10^{1}	0	0	9.55×10^{2}
F12	30	Mean	6.24×10^{-7}	7.45×10^{-3}	3.91×10^{-2}	8.53×10^{6}	7.37	2.42×10^{-2}	6.91×10^{-3}	1.05	7.71×10^{-1}	1.67×10^{-5}
		Std	1.16×10^{-6}	2.63×10^{-2}	1.88×10^{-2}	4.67×10^{7}	3.37	2.20×10^{-2}	2.63×10^{-2}	4.52×10^{-2}	7.15×10^{-2}	1.67×10^{-5}
	200	Mean	9.49×10^{-4}	3.80×10^{-1}	5.38×10^{-1}	2.17×10^{9}	4.93×10^{3}	7.15×10^{-2}	3.94×10^{1}	1.19	1.14	4.62×10^{-4}
		Std	1.56×10^{-3}	1.70×10^{-1}	4.58×10^{-2}	2.91×10^{8}	7.29×10^{3}	3.02×10^{-2}	1.98×10^{1}	2.79×10^{-2}	3.13×10^{-2}	3.36×10^{-4}
	500	Mean	1.44×10^{-3}	7.45×10^{-1}	7.46×10^{-1}	1.19×10^{10}	1.46×10^{6}	1.06×10^{-1}	2.38×10^{5}	1.19	1.17	9.62×10^{-4}
		Std	2.08×10^{-3}	2.14×10^{-1}	4.10×10^{-2}	6.36×10^{8}	9.21×10^{5}	5.63×10^{-2}	9.88×10^{4}	9.22×10^{-3}	8.84×10^{-3}	6.70×10^{-4}
	1000	Mean	1.00×10^{-3}	9.73×10^{-1}	1.20	3.06×10^{10}	1.11×10^{7}	9.41×10^{-2}	9.37×10^{6}	1.18	1.17	4.64×10^{-4}
		Std	2.08×10^{-3}	1.86×10^{-1}	2.17×10^{-1}	1.13×10^{9}	2.60×10^{6}	4.92×10^{-2}	2.32×10^{6}	3.00×10^{-3}	5.39×10^{-3}	4.04×10^{-4}

Table 3. *Cont.*

Function	*D*	Metric	IWHO	WHO	GWO	MFO	SSA	WOA	PSO	HSCAHS	DSCA	MALO
F13	30	Mean	4.35×10^{-5}	1.14×10^{-1}	5.87×10^{-1}	1.37×10^{7}	1.31×10^{1}	4.99×10^{-1}	4.46×10^{-3}	2.85	2.81	6.16×10^{-4}
		Std	1.38×10^{-4}	2.74×10^{-1}	2.68×10^{-1}	7.49×10^{7}	1.48×10^{1}	2.17×10^{-1}	8.92×10^{-3}	3.32×10^{-2}	4.64×10^{-2}	2.06×10^{-3}
	200	Mean	1.27×10^{-1}	1.79×10^{1}	1.69×10^{1}	4.49×10^{9}	1.33×10^{6}	6.22	5.70×10^{3}	1.99×10^{1}	1.99×10^{1}	9.64×10^{-2}
		Std	3.60×10^{-1}	3.27	6.35×10^{-1}	4.98×10^{8}	7.44×10^{5}	2.08	3.07×10^{3}	3.27×10^{-2}	5.31×10^{-2}	6.79×10^{-2}
	500	Mean	2.08×10^{-1}	5.62×10^{1}	5.05×10^{1}	2.23×10^{10}	3.59×10^{7}	1.91×10^{1}	4.14×10^{6}	4.99×10^{1}	5.00×10^{1}	4.49×10^{-1}
		Std	2.87×10^{-1}	1.15×10^{1}	1.29	1.21×10^{9}	9.80×10^{6}	5.58	1.17×10^{6}	2.52×10^{-2}	3.91×10^{-2}	2.67×10^{-1}
	1000	Mean	7.12×10^{-1}	1.15×10^{2}	1.25×10^{2}	5.54×10^{10}	1.54×10^{8}	3.96×10^{1}	8.03×10^{7}	9.99×10^{1}	1.00×10^{2}	1.05
		Std	1.02	1.00×10^{1}	1.40×10^{1}	1.68×10^{9}	2.46×10^{7}	1.18×10^{1}	1.17×10^{7}	2.82×10^{-2}	4.65×10^{-2}	6.62×10^{-1}
F14	2	Mean	9.98×10^{-1}	2.21	4.07	2.35	1.16	2.93	3.00	2.84	4.64×10^{-2}	1.63
		Std	1.30×10^{-16}	2.76	4.00	2.03	5.87×10^{-1}	3.07	2.54	4.52×10^{-1}	7.05×10^{-1}	9.20×10^{-1}
F15	4	Mean	4.48×10^{-4}	1.39×10^{-3}	2.48×10^{-3}	2.78×10^{-3}	3.48×10^{-3}	7.32×10^{-4}	4.94×10^{-3}	2.40×10^{-3}	1.31×10^{-3}	2.15×10^{-3}
		Std	2.56×10^{-4}	3.61×10^{-3}	6.07×10^{-3}	5.14×10^{-3}	6.74×10^{-3}	5.21×10^{-4}	7.86×10^{-3}	1.24×10^{-3}	3.22×10^{-4}	4.97×10^{-3}
F16	2	Mean	−1.03	−1.03	−1.03	−1.03	−1.03	−1.03	−1.03	−1.02	−1.03	−1.03
		Std	6.05×10^{-16}	4.97×10^{-16}	1.95×10^{-8}	6.78×10^{-16}	2.74×10^{-14}	9.52×10^{-10}	6.25×10^{-16}	5.56×10^{-3}	1.83×10^{-4}	1.28×10^{-13}
F17	2	Mean	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	6.76×10^{-1}	4.00×10^{-1}	3.98×10^{-1}
		Std	0	0	1.68×10^{-6}	0	9.46×10^{-15}	2.08×10^{-5}	0	2.05×10^{-1}	2.95×10^{-3}	2.20×10^{-14}
F18	2	Mean	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.01	3.01	3.00
		Std	1.82×10^{-15}	6.85×10^{-16}	3.02×10^{-5}	1.70×10^{-15}	2.22×10^{-13}	5.43×10^{-5}	1.33×10^{-15}	1.68×10^{-2}	8.68×10^{-3}	7.64×10^{-13}
F19	3	Mean	−3.86	−3.84	−3.86	−3.86	−3.86	−3.86	−3.86	−3.43	−3.83	−3.86
		Std	2.52×10^{-15}	1.41×10^{-1}	3.59×10^{-3}	2.71×10^{-15}	1.38×10^{-12}	7.94×10^{-3}	2.65×10^{-15}	3.26×10^{-1}	2.73×10^{-2}	6.35×10^{-13}
F20	6	Mean	−3.26	−3.25	−3.27	−3.24	−3.23	−3.20	−3.22	−1.78	−3.01	−3.25
		Std	6.05×10^{-2}	6.00×10^{-2}	7.38×10^{-2}	6.40×10^{-2}	6.34×10^{-2}	1.18×10^{-1}	1.07×10^{-1}	6.08×10^{-1}	9.02×10^{-2}	5.86×10^{-2}
F21	4	Mean	-1.02×10^{1}	−7.89	−9.32	−6.14	−7.06	−8.28	−6.71	-6.54×10^{-1}	−4.08	−6.86
		Std	5.49×10^{-15}	3.12	2.21	3.45	3.46	2.73	3.19	2.50×10^{-1}	9.62×10^{-1}	2.59
F22	4	Mean	-1.04×10^{1}	−8.51	-1.02×10^{1}	−7.36	−9.11	−8.18	−8.54	-7.37×10^{-1}	−4.48	−6.85
		Std	8.73×10^{-16}	3.22	9.63×10^{-1}	3.57	2.69	2.99	2.96	2.99×10^{-1}	2.34×10^{-1}	3.48
F23	4	Mean	-1.05×10^{1}	−7.33	-1.01×10^{1}	−7.95	−8.15	−7.31	−8.57	−1.01	−4.45	−6.78
		Std	2.06×10^{-15}	3.75	1.75	3.53	3.26	3.39	3.11	3.03×10^{-1}	1.08	3.64

Table 4. The Wilcoxon signed-rank test results between IWHO and other algorithms on classical test functions (F1–F23).

Function	D	IWHO vs. WHO	IWHO vs. GWO	IWHO vs. MFO	IWHO vs. SSA	IWHO vs. WOA	IWHO vs. PSO	IWHO vs. HSCAHS	IWHO vs. DSCA	IWHO vs. MALO
F1	30	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
	200	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
	500	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
	1000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F2	30	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
	200	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
	500	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
	1000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F3	30	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
	200	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
	500	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
	1000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F4	30	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
	200	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
	500	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
	1000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F5	30	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.22×10^{-4}	6.10×10^{-5}	6.10×10^{-5}	4.21×10^{-1}
	200	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.37×10^{-2}
	500	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.37×10^{-2}
	1000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.83×10^{-4}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.69×10^{-1}
F6	30	1.83×10^{-4}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	3.05×10^{-4}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
	200	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.22×10^{-4}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	5.54×10^{-2}
	500	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.16×10^{-3}
	1000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	2.52×10^{-1}
F7	30	1.22×10^{-4}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.22×10^{-4}	6.10×10^{-5}	6.10×10^{-5}	4.27×10^{-4}	6.10×10^{-5}
	200	3.05×10^{-4}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	2.62×10^{-3}	1.22×10^{-4}	4.79×10^{-2}
	500	8.54×10^{-4}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	8.54×10^{-4}	6.10×10^{-5}	6.71×10^{-3}	1.22×10^{-4}	1.53×10^{-3}
	1000	2.01×10^{-3}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	2.01×10^{-3}	6.10×10^{-5}	1.51×10^{-2}	1.22×10^{-4}	1.25×10^{-2}
F8	30	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.83×10^{-4}
	200	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
	500	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
	1000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}

Table 4. *Cont.*

Function	D	IWHO vs. WHO	IWHO vs. GWO	IWHO vs. MFO	IWHO vs. SSA	IWHO vs. WOA	IWHO vs. PSO	IWHO vs. HSCAHS	IWHO vs. DSCA	IWHO vs. MALO
F9	30	1.00	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.00	6.10×10^{-5}	1.00	1.00	6.10×10^{-5}
	200	1.00	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.00	6.10×10^{-5}	1.00	1.00	6.10×10^{-5}
	500	1.00	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.00	6.10×10^{-5}	1.00	1.00	6.10×10^{-5}
	1000	1.00	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.00	6.10×10^{-5}	1.00	1.00	6.10×10^{-5}
F10	30	1.25×10^{-1}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	9.77×10^{-4}	6.10×10^{-5}	1.00	1.00	6.10×10^{-5}
	200	1.25×10^{-1}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	3.91×10^{-3}	6.10×10^{-5}	1.00	1.00	6.10×10^{-5}
	500	6.25×10^{-2}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	7.81×10^{-3}	6.10×10^{-5}	1.00	1.00	6.10×10^{-5}
	1000	6.25×10^{-2}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.95×10^{-3}	6.10×10^{-5}	4.88×10^{-4}	1.00	6.10×10^{-5}
F11	30	1.00	1.25×10^{-1}	6.10×10^{-5}	6.10×10^{-5}	5.00×10^{-1}	6.10×10^{-5}	1.00	1.00	6.10×10^{-5}
	200	1.00	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.00	6.10×10^{-5}	1.00	1.00	6.10×10^{-5}
	500	1.00	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.00	6.10×10^{-5}	1.00	1.00	6.10×10^{-5}
	1000	1.00	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.00	6.10×10^{-5}	1.00	1.00	6.10×10^{-5}
F12	30	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	1.81×10^{-2}	6.10×10^{-5}	6.10×10^{-5}	1.83×10^{-4}
	200	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	7.62×10^{-1}
	500	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	8.04×10^{-1}
	1000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	3.03×10^{-1}
F13	30	3.36×10^{-3}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	5.37×10^{-3}	6.10×10^{-5}	6.10×10^{-5}	8.33×10^{-2}
	200	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	2.52×10^{-1}
	500	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	4.27×10^{-3}
	1000	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}	8.90×10^{-1}
F14	2	7.81×10^{-3}	6.10×10^{-5}	1.95×10^{-3}	1.95×10^{-3}	6.10×10^{-5}	9.77×10^{-4}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F15	4	8.36×10^{-3}	5.54×10^{-2}	3.66×10^{-4}	1.51×10^{-2}	2.29×10^{-1}	1.22×10^{-4}	6.10×10^{-5}	6.10×10^{-5}	4.27×10^{-3}
F16	2	1.00	6.10×10^{-5}	1.00	1.22×10^{-4}	6.10×10^{-5}	1.00	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F17	2	1.00	6.10×10^{-5}	1.00	4.88×10^{-4}	6.10×10^{-5}	1.00	6.10×10^{-5}	6.10×10^{-5}	1.22×10^{-4}
F18	2	1.95×10^{-3}	6.10×10^{-5}	7.54×10^{-1}	6.10×10^{-5}	6.10×10^{-5}	3.07×10^{-1}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F19	3	1.00	6.10×10^{-5}	1.00	6.10×10^{-5}	6.10×10^{-5}	1.00	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F20	6	7.66×10^{-1}	2.56×10^{-2}	1.56×10^{-2}	8.54×10^{-4}	2.52×10^{-1}	6.27×10^{-1}	6.10×10^{-5}	6.10×10^{-5}	2.01×10^{-3}
F21	4	3.13×10^{-2}	6.10×10^{-5}	3.13×10^{-2}	6.10×10^{-5}	6.10×10^{-5}	7.81×10^{-3}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F22	4	6.25×10^{-2}	6.10×10^{-5}	6.25×10^{-2}	6.10×10^{-5}	6.10×10^{-5}	3.91×10^{-3}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
F23	4	3.13×10^{-2}	6.10×10^{-5}	1.95×10^{-3}	6.10×10^{-5}	6.10×10^{-5}	1.56×10^{-2}	6.10×10^{-5}	6.10×10^{-5}	6.10×10^{-5}
Overall (+/=/−)		45/17/0	59/3/0	57/5/0	62/0/0	52/10/0	57/5/0	47/11/4	50/12/0	45/12/5

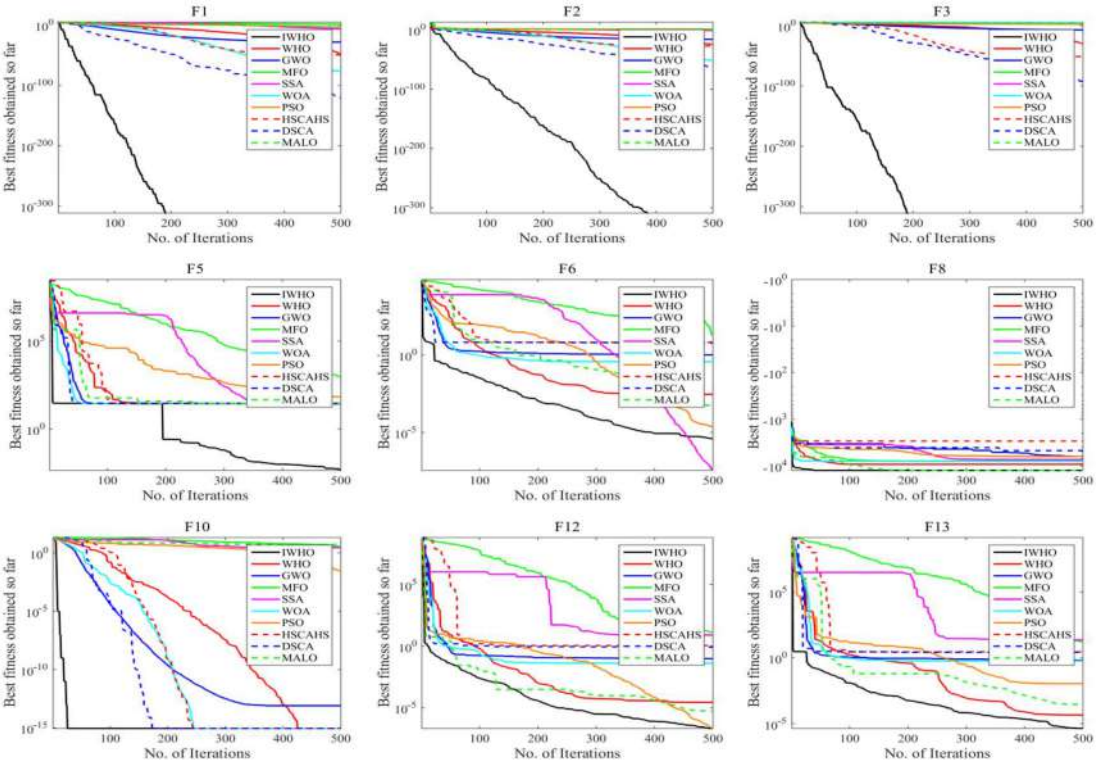

Figure 3. Convergence curves for the optimization algorithms on test functions (F1, F2, F3, F5, F6, F8, F10, F12, F13) with $D = 30$.

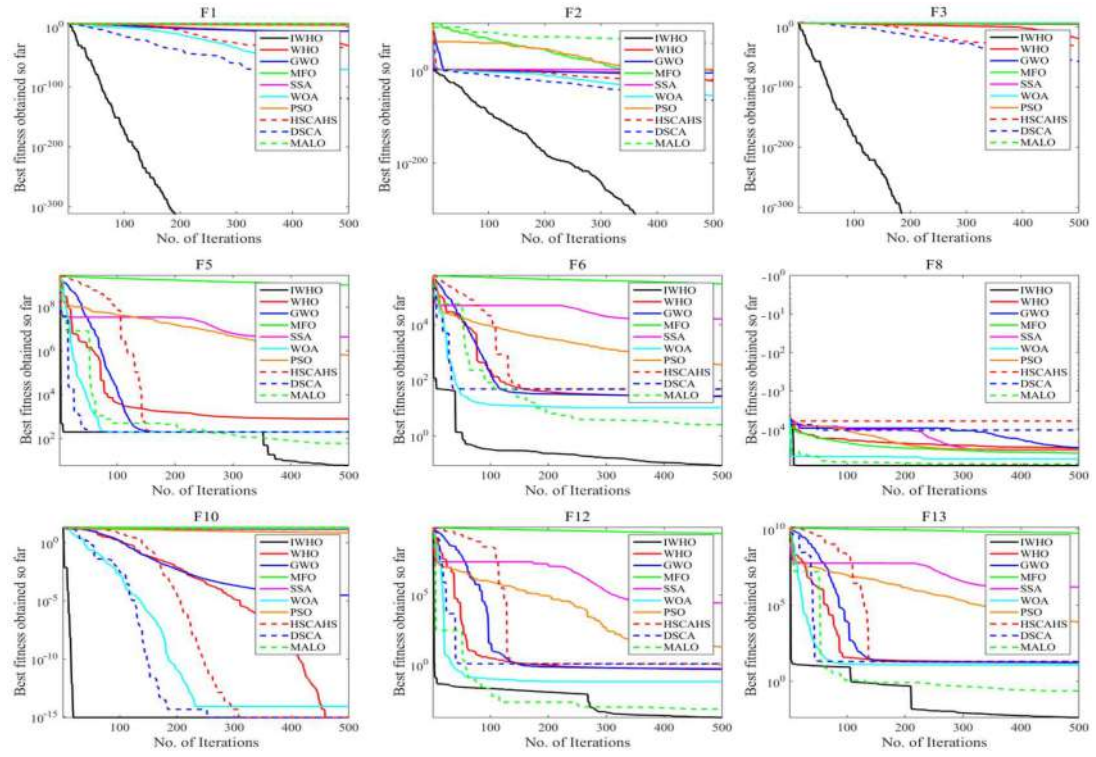

Figure 4. Convergence curves for the optimization algorithms on test functions (F1, F2, F3, F5, F6, F8, F10, F12, F13) with $D = 200$.

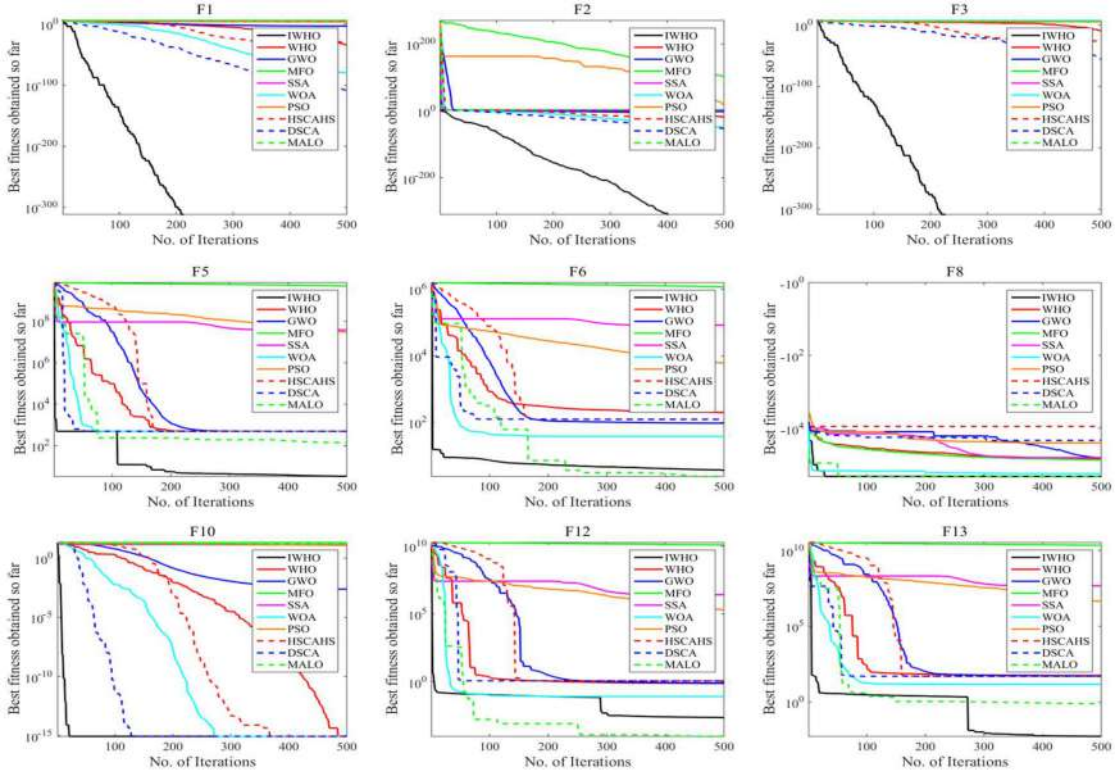

Figure 5. Convergence curves for the optimization algorithms on test functions (F1, F2, F3, F5, F6, F8, F10, F12, F13) with D = 500.

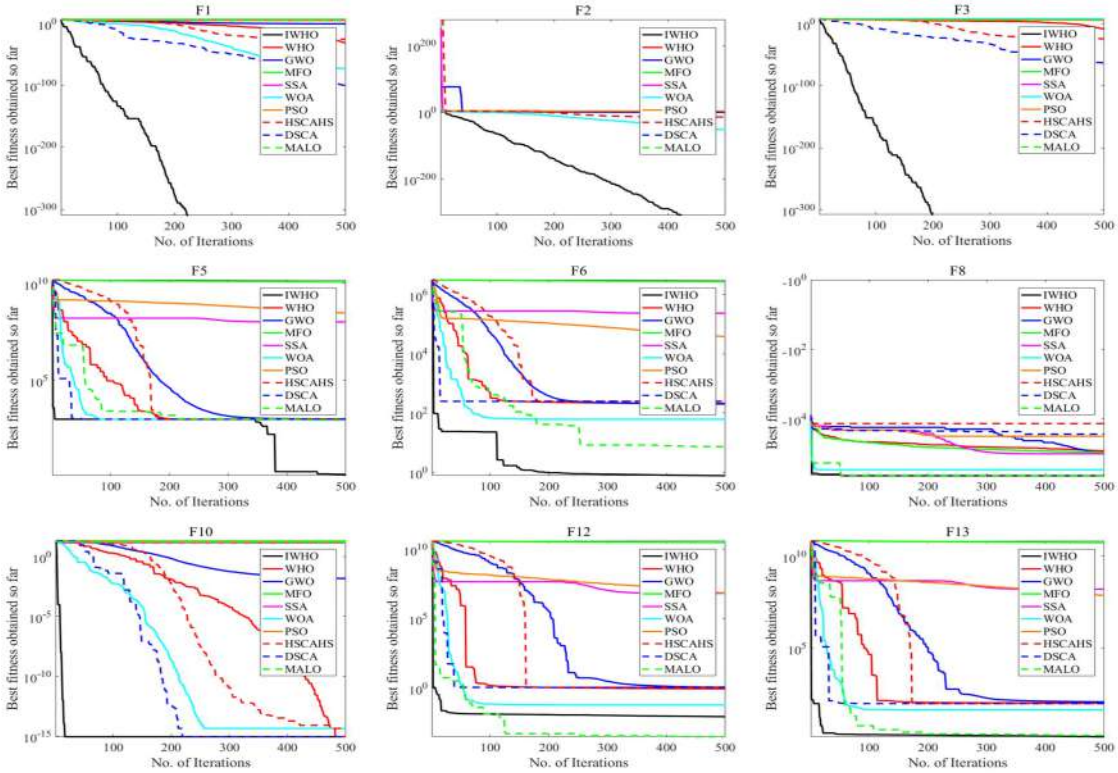

Figure 6. Convergence curves for the optimization algorithms on test functions (F1, F2, F3, F5, F6, F8, F10, F12, F13) with D = 1000.

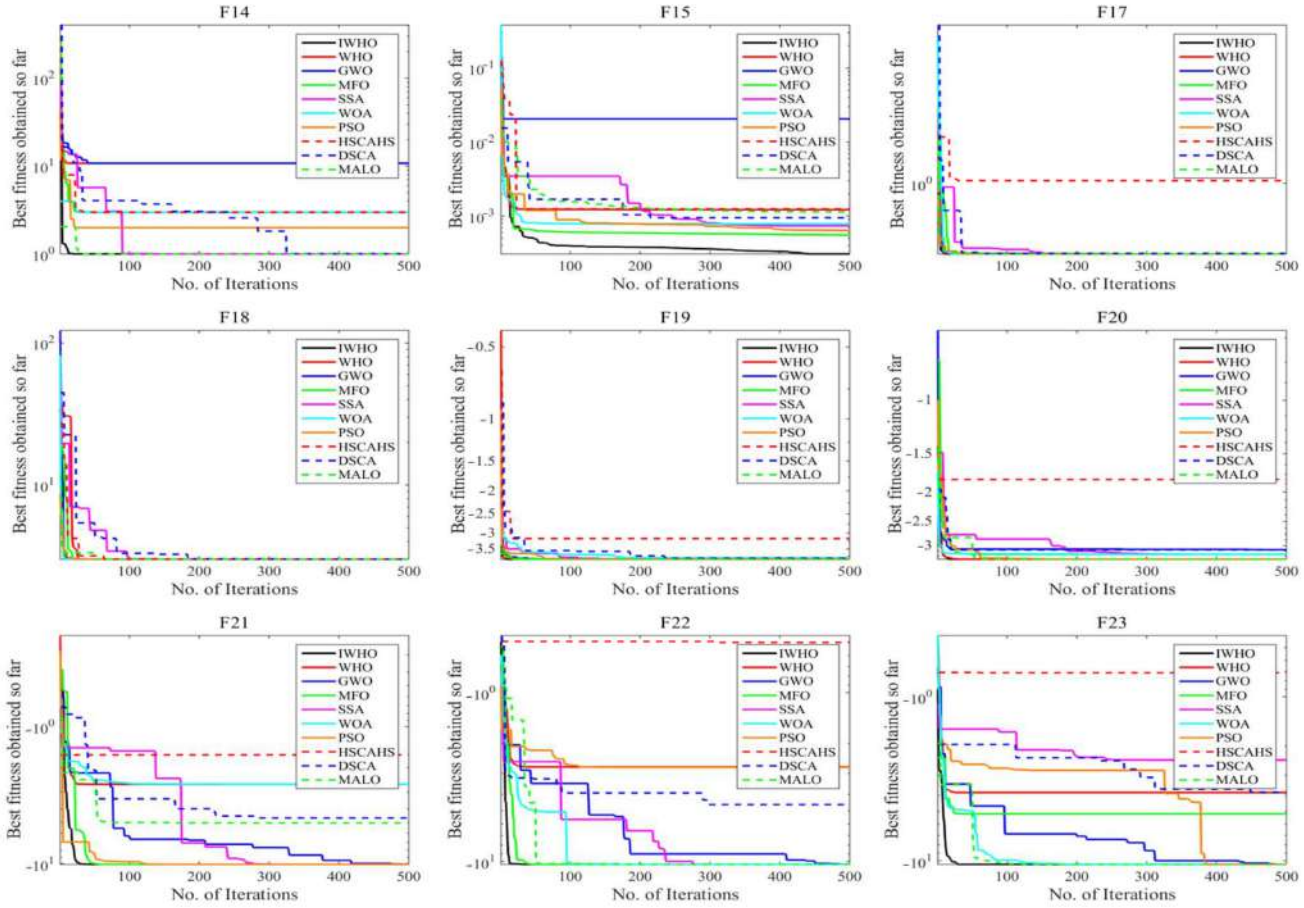

Figure 7. Convergence curves for the optimization algorithms on fixed-dimension test functions (F14, F15, F17–F23).

4.3. Statistical Analysis of Performance for the CEC 2021 Test Suite

In this subsection, the performance of IWHO is tested using ten mathematical functions from IEEE CEC 2021, and it is compared using the same comparative metaheuristics algorithms.

The results of CEC2021 test functions are given in Table 5 in terms of mean and standard deviation. Table 5 also shows the results of Friedman ranking test [63]. It can be seen that the proposed IWHO is ranked first in CEC_02 and CEC_06 and the second-best in the other two functions (CEC_05 and CEC_08). Overall, it is also ranked first.

Table 5. Comparison of optimization results on CEC2021 test functions between IWHO and other optimization algorithms (CEC_01−CEC_10).

Function	Metric	IWHO	WHO	GWO	MFO	SSA	WOA	PSO	HSCAHS	DSCA	MALO
CEC_01	Mean	2.64×10^{3}	4.54×10^{3}	8.07×10^{8}	2.13×10^{9}	2.41×10^{3}	1.44×10^{9}	1.58×10^{3}	3.25×10^{10}	1.40×10^{10}	1.70×10^{3}
	Std	3.02×10^{3}	4.17×10^{3}	6.80×10^{8}	1.62×10^{9}	2.82×10^{3}	1.03×10^{9}	1.94×10^{3}	2.42×10^{9}	2.58×10^{9}	1.99×10^{3}
	Ranking	4	5	6	8	3	7	1	10	9	2
CEC_02	Mean	2.60×10^{3}	2.76×10^{3}	2.74×10^{3}	3.28×10^{3}	3.52×10^{3}	4.34×10^{3}	3.38×10^{3}	6.29×10^{3}	5.76×10^{3}	3.36×10^{3}
	Std	4.49×10^{2}	4.21×10^{2}	7.65×10^{2}	7.86×10^{2}	5.76×10^{2}	6.19×10^{2}	3.65×10^{2}	2.98×10^{2}	2.33×10^{2}	6.28×10^{2}
	Ranking	1	3	2	4	7	8	6	10	9	5
CEC_03	Mean	7.93×10^{2}	7.92×10^{2}	7.94×10^{2}	8.37×10^{2}	8.21×10^{2}	9.68×10^{2}	7.85×10^{2}	1.06×10^{3}	9.86×10^{2}	8.61×10^{2}
	Std	2.45×10^{1}	2.89×10^{1}	2.66×10^{1}	7.11×10^{1}	4.42×10^{1}	4.63×10^{1}	1.86×10^{1}	1.78×10^{1}	1.82×10^{1}	5.06×10^{1}
	Ranking	3	2	4	6	5	8	1	10	9	7
CEC_04	Mean	1.91×10^{3}	1.91×10^{3}	2.02×10^{3}	1.05×10^{4}	1.91×10^{3}	2.80×10^{3}	1.90×10^{3}	4.20×10^{5}	8.47×10^{4}	1.91×10^{3}
	Std	3.44	4.80	2.92×10^{2}	1.22×10^{4}	2.24	1.81×10^{3}	1.60	2.13×10^{5}	4.36×10^{4}	3.19
	Ranking	3.5	3.5	6	8	3.5	7	1	10	9	3.5
CEC_05	Mean	2.26×10^{5}	2.76×10^{5}	8.21×10^{5}	2.87×10^{6}	6.26×10^{5}	2.43×10^{6}	1.92×10^{5}	6.72×10^{6}	4.43×10^{6}	2.92×10^{5}
	Std	1.52×10^{5}	2.00×10^{5}	6.45×10^{5}	4.73×10^{6}	4.90×10^{5}	1.20×10^{6}	1.35×10^{5}	2.02×10^{6}	2.22×10^{6}	2.27×10^{5}
	Ranking	2	3	6	8	5	7	1	10	9	4
CEC_06	Mean	1.86×10^{3}	1.99×10^{3}	2.01×10^{3}	2.04×10^{3}	2.26×10^{3}	2.65×10^{3}	2.19×10^{3}	3.82×10^{3}	2.75×10^{3}	2.41×10^{3}
	Std	1.53×10^{2}	1.93×10^{2}	1.69×10^{2}	2.10×10^{2}	2.30×10^{2}	3.11×10^{2}	2.35×10^{2}	2.69×10^{2}	2.10×10^{2}	2.74×10^{2}
	Ranking	1	2	3	4	6	8	5	10	9	7
CEC_07	Mean	1.43×10^{5}	9.90×10^{4}	3.30×10^{5}	1.07×10^{6}	2.39×10^{5}	1.34×10^{6}	1.07×10^{5}	2.78×10^{6}	2.23×10^{6}	1.30×10^{5}
	Std	1.15×10^{5}	1.10×10^{5}	4.35×10^{5}	1.59×10^{6}	2.27×10^{5}	1.00×10^{6}	9.74×10^{4}	1.54×10^{6}	1.16×10^{6}	1.18×10^{5}
	Ranking	4	1	6	7	5	8	2	10	9	3
CEC_08	Mean	2.36×10^{3}	2.80×10^{3}	3.02×10^{3}	4.07×10^{3}	2.93×10^{3}	4.21×10^{3}	3.64×10^{3}	5.91×10^{3}	6.04×10^{3}	2.30×10^{3}
	Std	3.38×10^{2}	1.03×10^{3}	9.10×10^{2}	1.32×10^{3}	1.31×10^{3}	1.92×10^{3}	1.49×10^{3}	4.45×10^{2}	8.07×10^{2}	1.15×10^{1}
	Ranking	2	3	5	7	4	8	6	9	10	1
CEC_09	Mean	2.88×10^{3}	2.87×10^{3}	2.87×10^{3}	2.89×10^{3}	2.86×10^{3}	3.04×10^{3}	3.06×10^{3}	3.47×10^{3}	3.02×10^{3}	2.90×10^{3}
	Std	3.78×10^{1}	3.22×10^{1}	3.95×10^{1}	3.40×10^{1}	2.14×10^{1}	7.19×10^{1}	8.00×10^{1}	9.40×10^{1}	1.99×10^{1}	2.36×10^{1}
	Ranking	4	2.5	2.5	5	1	8	9	10	7	6
CEC_10	Mean	2.96×10^{3}	2.97×10^{3}	3.00×10^{3}	3.04×10^{3}	2.95×10^{3}	3.13×10^{3}	2.95×10^{3}	5.29×10^{3}	3.84×10^{3}	2.97×10^{3}
	Std	3.77×10^{1}	3.22×10^{1}	3.95×10^{1}	1.56×10^{2}	3.29×10^{1}	7.21×10^{1}	2.97×10^{1}	2.94×10^{2}	2.42×10^{2}	2.64×10^{1}
	Ranking	3	4.5	6	7	1.5	8	1.5	10	9	4.5
Overall Ranking		1	2	6	7	4	8	3	10	9	5

Figure 8 shows the box plots of the introduced algorithm and other compared ones using nine functions (CEC_01–CEC_09). It can be seen that the improved algorithm has better stability in these functions. Further, Figure 9 shows different exploration and exploitation phases. The percentages of exploration and exploitation phases are calculated based on the dispersion of all searching individuals, which exhibits the flexibility of improved algorithms during the optimization process.

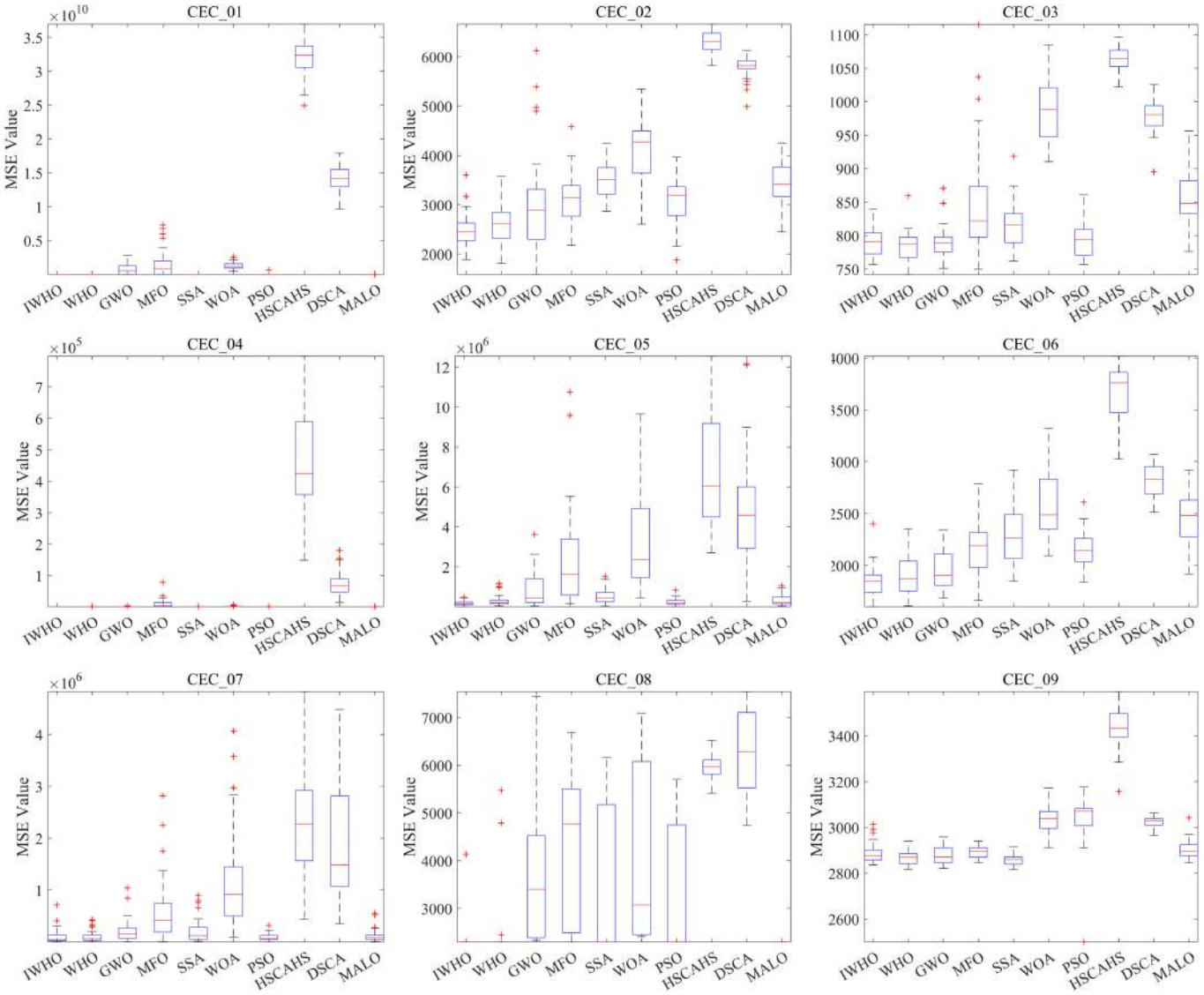

Figure 8. Box plots of CEC2021 test functions (CEC_01–CEC_09).

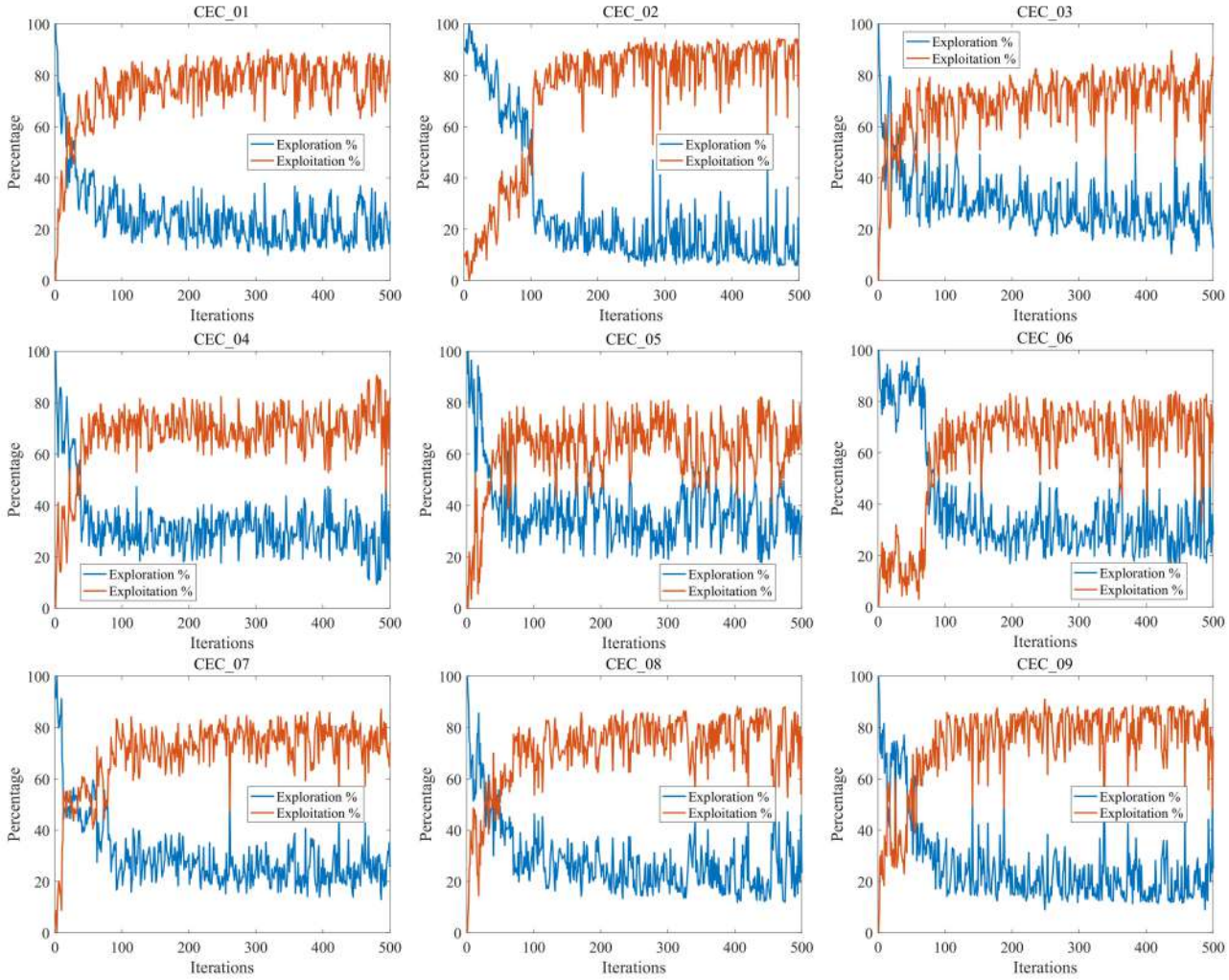

Figure 9. Exploration and exploitation phases for IWHO on CEC2021 test functions (CEC_01–CEC_09).

5. Real-World Applications

In this section, several real-world engineering design problems with constraints are utilized to assess IWHO's significance and effectiveness. IWHO has been evaluated using five engineering design problems: welded beam, tension/compression spring, three-bar truss, car crashworthiness, and speed reducer design problems.

5.1. Welded Beam Design Problem

The first engineering problem used to evaluate IWHO is the welded beam design, a common and well-known problem [64], as shown in Figure 10. Welded beam aims to select the optimal variables to decrease the entire weld beam manufacturing cost subject to:

- Welded thickness (h)
- Bar length (l)
- Bar height (t)
- Bar thickness (b)

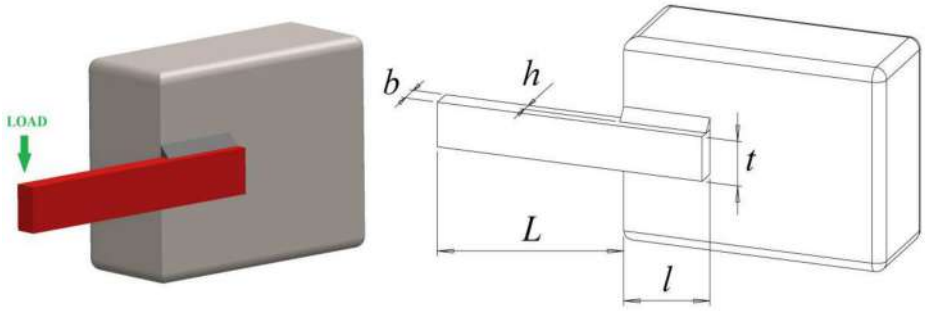

Figure 10. The welded beam design problem: 3D model diagram (**left**) and structural parameters (**right**).

The mathematical equations of this problem are shown below:
Consider:

$$\vec{x} = [x_1,\ x_2,\ x_3,\ x_4] = [h,\ l,\ t,\ b]$$

Minimize:

$$f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$$

Subject to:

$$\begin{gathered} g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \leq 0 \\ g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{\max} \leq 0 \\ g_3(\vec{x}) = \delta(\vec{x}) - \delta_{\max} \leq 0 \\ g_4(\vec{x}) = x_1 - x_4 \leq 0 \\ g_5(\vec{x}) = P - P_C(\vec{x}) \leq 0 \\ g_6(\vec{x}) = 0.125 - x_1 \leq 0 \\ g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5 \leq 0 \end{gathered}$$

where:

$$\begin{gathered} \tau(\vec{x}) = \sqrt{(\tau\prime)^2 + 2\tau\prime\tau'' \frac{x_2}{2R} + (\tau'')^2} \\ \tau\prime = \frac{P}{\sqrt{2}x_1x_2},\ \tau'' = \frac{MR}{J},\ M = P(L + \frac{x_2}{2}) \\ R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2} \\ J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2\right]\right\} \\ \sigma(\vec{x}) = \frac{6PL}{x_3^2x_4},\ \delta(\vec{x}) = \frac{6PL^3}{Ex_3^3x_4} \\ P_C(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \end{gathered}$$

Range of variables:

$$\begin{gathered} P = 6000 \text{ lb},\ L = 14 \text{ in},\ E = 30 \times 10^6 \text{ psi},\ G = 12 \times 10^6 \text{ psi} \\ \tau_{\max} = 13600 \text{ psi},\ \sigma_{\max} = 30000 \text{ psi},\ \delta_{\max} = 0.25 \text{ in} \end{gathered}$$

IWHO has been compared to the original WHO [49], GWO [50], MFO [51], WOA [53], SCA [65], ALO [66], DA [67], and LSHADE [68]. From Table 6, it's obvious that IWHO achieves the best results compared to other algorithms. It achieves [*h*, *l*, *t*, *b*] = [0.2057, 3.2530, 9.0366, 0.2057] with fitness value 1.6952.

Table 6. Optimization results for the welded beam design problem.

Algorithm	Optimal Values for Variables				Minimum Cost
	h	*l*	*t*	*b*	
IWHO	0.2057	3.2530	9.0366	0.2057	1.6952
WHO [49]	0.2058	3.2514	9.0370	0.2058	1.6958
GWO [50]	0.202369	3.544214	9.048210	0.205723	1.728712
MFO [51]	0.302546	2.662619	7.262456	0.318524	1.726186
WOA [53]	0.328517	2.465267	7.046271	0.426580	1.878023
SCA [65]	0.141883	5.225674	10.00000	0.211640	1.858033
ALO [66]	0.125130	9.784118	6.522153	0.394937	1.785364
DA [67]	0.206905	3.314990	9.397212	0.206905	1.749248
LSHADE [68]	0.20573	3.4705	9.0366	0.20573	1.724852

5.2. Tension/Compression Spring Design Problem

The second constrained problem is the tension/compression spring problem [69], the goal of which is to select the optimal parameters to minimize the price of manufacturing a spring, as shown in Figure 11. This problem has three variables:

- Diameter of wire (d)
- Coil diameter (D)
- Active coils number (N)

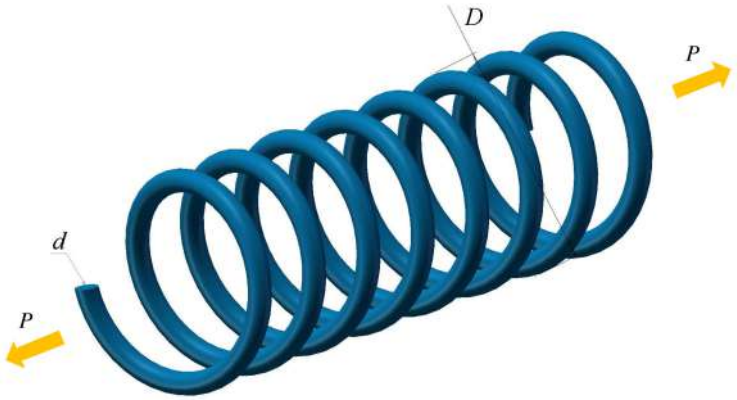

Figure 11. Tension/compression spring design problem (3D model diagram and structural parameters).

The mathematical description of this problem is given below:
Consider:

$$\vec{x} = [x_1,\ x_2,\ x_3,\ x_4] = [d,\ D,\ N]$$

Minimize:

$$f(\vec{x}) = (x_3 + 2)x_2x_1^2$$

Subject to:

$$g_1(\vec{x}) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0$$
$$g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \leq 0$$
$$g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0$$
$$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

Range of variables:

$$0.05 \leq x_1 \leq 2.00$$
$$0.25 \leq x_2 \leq 1.30$$
$$2.00 \leq x_3 \leq 15.00$$

Table 7 shows the results of the proposed algorithm compared to AO [70], HHO [71], SSA [52], WOA [53], GWO [50], PSO [15], MVO [72], GA [29], and HS [73]. As shown in Table 8, IWHO clearly ranked first, with [*d*, *D*, *N*] = [0.0517, 0.4155, 7.1564] and a fitness value of 0.0102.

Table 7. Optimization results for the tension/compression spring design problem.

Algorithm	Optimal Values for Variables			Minimum Weight
	d	*D*	*N*	
IWHO	0.0517	0.4155	7.1564	0.0102
AO [70]	0.0502439	0.35262	10.5425	0.011165
HHO [71]	0.051796393	0.359305355	11.138859	0.012665443
SSA [52]	0.051207	0.345215	12.004032	0.0126763
WOA [53]	0.051207	0.345215	12.004032	0.0126763
GWO [50]	0.05169	0.356737	11.28885	0.012666
PSO [15]	0.051728	0.357644	11.244543	0.0126747
MVO [72]	0.05251	0.37602	10.33513	0.012790
GA [30]	0.051480	0.351661	11.632201	0.01270478
HS [73]	0.051154	0.349871	12.076432	0.0126706

5.3. Three-Bar Truss Design Problem

This engineering problem is a non-linear optimization problem introduced by Nowcki to minimize bar structure burdens [74]. Three-bar truss has two variables, as shown in Figure 12. The mathematical description of this problem is given below:

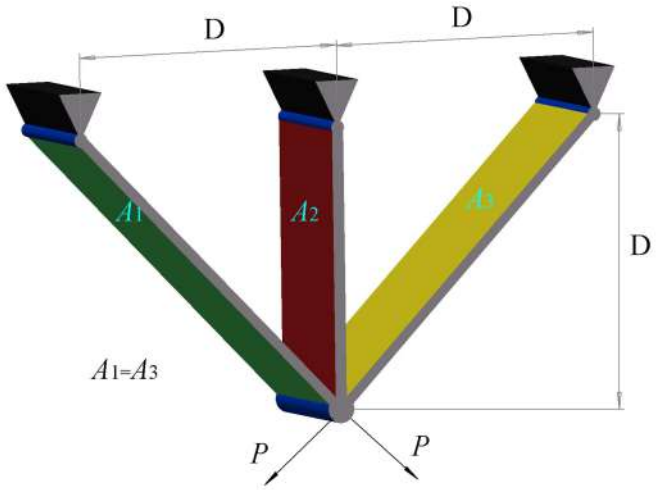

Figure 12. Three-bar truss design problem (3D model diagram and structural parameters).

Consider:

$$\vec{x} = [x_1,\ x_2] = [A_1,\ A_2]$$

Minimize:

$$f(\vec{x}) = (2\sqrt{2}x_1 + x_2)l$$

Subject to:

$$g_1(\vec{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0$$
$$g_2(\vec{x}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0$$
$$g_3(\vec{x}) = \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \leq 0$$

where:

$$l = 100\ \text{cm},\ P = 2\ \text{KN/cm}^2,\ \sigma = 2\ \text{KN/cm}^2$$

Range of variables:

$$0 \leq x_1,\ x_2 \leq 1$$

The results of this problem are given in Table 8, comparing IWHO with basic WHO [49], HHO [71], SSA [52], AOA [75], MVO [72], MFO [51], and GOA [18]. IWHO achieves the best solution $[x_1, x_2]$ = [0.7884, 0.4081] and its objective value is 263.8523.

Table 8. Optimization results for the three-bar truss design problem.

Algorithm	Optimal Values for Variables		Optimum Weight
	x_1	x_2	
IWHO	0.7884	0.4081	263.8523
WHO [49]	0.7980	0.3816	263.9181
HHO [71]	0.788662816	0.408283133832900	263.8958434
SSA [52]	0.78866541	0.408275784	263.89584
AOA [75]	0.79369	0.39426	263.9154
MVO [72]	0.78860276	0.408453070000000	263.8958499
MFO [51]	0.788244771	0.409466905784741	263.8959797
GOA [18]	0.788897555578973	0.407619570115153	263.895881496069

5.4. Car Crashworthiness Design Problem

The car crashworthiness problem was first introduced by Gu et al. [76]. This engineering problem has 10 constraints and 11 design variables. The mathematical description and 3D model diagram of this problem can be seen in [77].

The results of this problem are given in Table 9, where IWHO is compared to GWO [50], GOA [18], PSO [15], MFO [51], SSA [52], DSCA [55], MALO [56]. From this table, we can conclude that IWHO achieves the best results $[x_1 - x_{11}]$ = [0.5000, 1.1186, 0.5000, 1.2982, 0.5000, 1.5000, 0.5000, 0.3450, 0.3450, −19.1594, 0.0020] with fitness value 22.8427.

Table 9. Optimization results in car crashworthiness design problem.

Algorithm	IWHO	GWO [50]	GOA [18]	PSO [15]	MFO [51]	SSA [52]	DSCA [55]	MALO [56]
x_1	0.5000	0.5000	0.50000	0.5000	0.5000	0.5000	0.5000	0.5000
x_2	1.1186	1.0268	1.11670	1.1327	1.0946	1.1221	0.9765	1.2023
x_3	0.5000	0.5929	0.50000	0.5000	0.5000	0.5062	1.3381	0.5000
x_4	1.2982	1.3772	1.30208	1.2781	1.3461	1.3170	0.5000	1.2833
x_5	0.5000	0.5000	0.50000	0.5000	0.5000	0.5000	0.5214	0.5000
x_6	1.5000	1.5000	1.50000	1.5000	1.5000	1.5000	1.5000	1.1728
x_7	0.5000	0.5000	0.50000	0.5000	0.5000	0.5000	0.6263	0.5000
x_8	0.3450	0.3450	0.34500	0.3450	0.3450	0.3136	0.1920	0.3347
x_9	0.3450	0.3450	0.19200	0.3450	0.3450	0.2115	0.3450	0.3096
x_{10}	−19.1594	−30.0000	−19.54935	−16.6449	−23.4599	−21.9139	−30.0000	−5.3953
x_{11}	0.0020	0.1523	−0.00431	0.1076	0.0014	2.2405	7.8909	7.3999
Optimal value	22.8427	23.1976	22.84474	22.8556	22.8744	22.9855	24.9261	23.3406

5.5. Speed Reducer Design Problem

The last constrained engineering problem is the speed reducer design problem [75], as shown in Figure 13, which aims to find the minimum weight of the speed reducer. This problem has seven variables. The mathematical formulation of this problem is shown below:

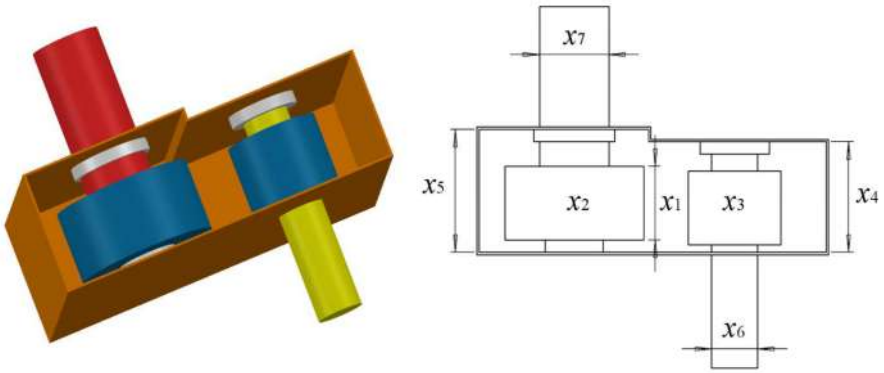

Figure 13. The speed reducer design problem: 3D model diagram (**left**) and structural parameters (**right**).

Minimize:

$$f(\vec{x}) = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3)$$

Subject to:

$$g_1\left(\vec{x}\right) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0$$
$$g_2\left(\vec{x}\right) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0$$
$$g_3\left(\vec{x}\right) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \leq 0$$
$$g_4\left(\vec{x}\right) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \leq 0$$
$$g_5\left(\vec{x}\right) = \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9\times10^6}}{110.0x_6^3} - 1 \leq 0$$
$$g_6\left(\vec{x}\right) = \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 157.5\times10^6}}{85.0x_6^3} - 1 \leq 0$$
$$g_7\left(\vec{x}\right) = \frac{x_2x_3}{40} - 1 \leq 0$$
$$g_8\left(\vec{x}\right) = \frac{5x_2}{x_1} - 1 \leq 0$$
$$g_9\left(\vec{x}\right) = \frac{x_1}{12x_2} - 1 \leq 0$$
$$g_{10}\left(\vec{x}\right) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0$$
$$g_{11}\left(\vec{x}\right) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0$$

Range of variables:

$$2.6 \leq x_1 \leq 3.6$$
$$0.7 \leq x_2 \leq 0.8$$
$$17 \leq x_3 \leq 28$$
$$7.3 \leq x_4 \leq 8.3$$
$$7.8 \leq x_5 \leq 8.3$$
$$2.9 \leq x_6 \leq 3.9$$
$$5.0 \leq x_7 \leq 5.5$$

IWHO is compared to AO [70], PSO [15], AOA [75], MFO [51], GA [30], SCA [65], HS [73], and FA [78]. As listed in Table 10, the proposed algorithm clearly achieves the best results: $[x_1 - x_7]$ = [3.4976, 0.7, 17, 7.3, 7.8, 3.3501, 5.2855] with fitness value 2995.43736.

Table 10. Optimization results for the speed reducer design problem.

Algorithm	Optimal Values for Variables							Optimum Weight
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
IWHO	3.4976	0.7	17	7.3	7.8	3.3501	5.2855	2995.43736
AO [70]	3.5021	0.7	17	7.3099	7.7476	3.3641	5.2994	3007.7328
PSO [15]	3.5001	0.7	17.0002	7.5177	7.7832	3.3508	5.2867	3145.922
AOA [75]	3.50384	0.7	17	7.3	7.72933	3.35649	5.2867	2997.9157
MFO [51]	3.49745	0.7	17	7.82775	7.71245	3.35178	5.28635	2998.9408
GA [29]	3.51025	0.7	17	8.35	7.8	3.36220	5.28772	3067.561
SCA [65]	3.50875	0.7	17	7.3	7.8	3.46102	5.28921	3030.563
HS [73]	3.52012	0.7	17	8.37	7.8	3.36697	5.28871	3029.002
FA [78]	3.50749	0.7001	17	7.71967	8.08085	3.35151	5.28705	3010.13749

6. Conclusions

In this paper, an improved wild horse optimizer (IWHO) is proposed for solving global optimization problems. The basic WHO is enhanced by integrating three improvements: random running strategy (RRS), dynamic inertia weight strategy (DIWS), and competition for waterhole mechanism (CWHM), which were inspired by the behavior of wild horses. RRS is utilized to enhance the global search capability and balance exploration and exploitation, while DIWS and CWHM are applied to increase the quality of the optimal solution. To evaluate the performance of the proposed IWHO, twenty-three classical benchmark functions with various types and dimensions, ten CEC 2021 test functions, and five real-world optimization problems are employed for testing. Meanwhile, nine well-known algorithms are used for comparison: basic wild horse optimizer (WHO), grey wolf optimizer (GWO), moth–flame optimization (MFO), salp swarm algorithm (SSA), whale optimization algorithm (WOA), particle swarm optimizer (PSO), hybridizing sine–cosine algorithm with harmony search (HSCAHS), dynamic sine–cosine algorithm (DSCA), and modified ant lion optimizer (MALO). The numerical and statistical results indicate that IWHO outperforms other methods for most test functions, showing that the improvements applied to WHO are successful. In addition, the results of engineering design problems verify the applicability of the proposed IWHO.

In the future, IWHO may be applied to high-dimensional, real-world problems, such as multilayer perceptron training for solving classification tasks. Additionally, the IWHO can be hybridized with other metaheuristic algorithms to further improve its optimization capability and solve more optimization problems, such as feature selection, multilevel thresholding image segmentation, and parameter optimization.

Author Contributions: Conceptualization, R.Z.; methodology, R.Z. and H.-M.J.; software, S.W.; validation, D.W.; writing—original draft preparation, A.G.H. and R.Z.; writing—review and editing, L.A. and H.-M.J.; funding acquisition, R.Z. and H.-M.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by Sanming University, which introduces high-level talent to start scientific research; funding support project (21YG01, 20YG14), Sanming University National Natural Science Foundation Breeding Project (PYT2103, PYT2105), Fujian Natural Science Foundation Project (2021J011128), Guiding Science and Technology Projects in Sanming City (2021-S-8), Educational Research Projects of Young and Middle-Aged Teachers in Fujian Province (JAT200618), Scientific Research and Development Fund of Sanming University (B202009), and by Open Research Fund Program of Fujian Provincial Key Laboratory of Agriculture Internet of Things Application (ZD2101).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors would like to thank the support of Fujian Key Lab of Agriculture IoT Application and IoT Application Engineering Research Center of Fujian Province Colleges and Universities, and also the anonymous reviewers and the editor for their careful reviews and constructive suggestions to help us improve the quality of this paper.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Hassanien, A.E.; Emary, E. *Swarm Intelligence: Principles, Advances, and Applications*; CRC Press: Boca Raton, FL, USA, 2018; pp. 93–119.
2. Hussien, A.G.; Hassanien, A.E.; Houssein, E.H.; Amin, M.; Azar, A.T. New binary whale optimization algorithm for discrete optimization problems. *Eng. Optimiz.* **2020**, *52*, 945–959. [CrossRef]
3. Yang, X.S. *Engineering Optimization: An Introduction with Metaheuristic Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2010.
4. Jamil, M.; Yang, X.S. A literature survey of benchmark functions for global optimization problems. *Int. J. Math. Model. Numer. Optim.* **2013**, *4*, 150–194.
5. Fathi, H.; AlSalman, H.; Gumaei, A.; Manhrawy, I.I.M.; Hussien, A.G.; El-Kafrawy, P. An Efficient Cancer Classification Model Using Microarray and High-Dimensional Data. *Comput. Intel. Neurosc.* **2021**, *2021*, 7231126. [CrossRef] [PubMed]
6. Mousavi-Avval, S.H.; Rafiee, S.; Sharifi, M.; Hosseinpour, S.; Notarnicola, B.; Tassielli, G.; Renzulli, P.A. Application of multi-objective genetic algorithms for optimization of energy, economics and environmental life cycle assessment in oilseed production. *J. Clean. Prod.* **2017**, *140*, 804–815. [CrossRef]
7. Abualigah, L.; Gandomi, A.H.; Elaziz, M.A.; Hussien, A.G.; Khasawneh, A.M.; Alshinwan, M.; Houssein, E.H. Nature-inspired optimization algorithms for text document clustering—A comprehensive analysis. *Algorithms* **2020**, *13*, 345. [CrossRef]
8. Shamir, J.; Rosen, J.; Mahlab, U.; Caulfield, H.J. Optimization methods for pattern recognition. *Int. Soc. Opt. Eng.* **1992**, *40*, 2–24.
9. Houssein, E.H.; Amin, M.; Hassanien, A.G.; Houssein, A.E. Swarming behaviour of salps algorithm for predicting chemical compound activities. In Proceedings of the 8th IEEE International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 5–7 December 2017; pp. 315–320.
10. Chou, J.S.; Pham, A.D. Nature-inspired metaheuristic optimization in least squares support vector regression for obtaining bridge scour information. *Inform. Sci.* **2017**, *399*, 64–80. [CrossRef]
11. Hussien, A.G.; Hassanien, A.E.; Houssein, E.H.; Bhattacharyya, S.; Amin, M. S-shaped Binary Whale Optimization Algorithm for Feature Selection. In *Recent Trends in Signal and Image Processing*; Springer: Singapore, 2019; Volume 727, pp. 79–87.
12. Hussien, A.G.; Houssein, E.H.; Hassanien, A.E. A binary whale optimization algorithm with hyperbolic tangent fitness function for feature selection. In Proceedings of the 8th IEEE International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 5–7 December 2017; pp. 166–172.
13. Besnassi, M.; Neggaz, N.; Benyettou, A. Face detection based on evolutionary Haar filter. *Pattern Anal. Appl.* **2020**, *23*, 309–330. [CrossRef]
14. Wang, Z.Y.; Xing, H.L.; Li, T.R.; Yang, Y.; Qu, R.; Pan, Y. A modified ant colony optimization algorithm for network coding resource minimization. *IEEE T. Evolut. Comput.* **2016**, *20*, 325–342. [CrossRef]
15. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
16. Dorigo, M.; Gambardella, L.M. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE T. Evolut. Comput.* **1997**, *1*, 53–66. [CrossRef]
17. Yang, X.S.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 210–214.
18. Saremi, S.; Mirjalili, S.; Lewis, A. Grasshopper optimisation algorithm: Theory and application. *Adv. Eng. Softw.* **2017**, *105*, 30–47. [CrossRef]
19. Li, M.D.; Zhao, H.; Weng, X.W.; Han, T. A novel nature-inspired algorithm for optimization: Virus colony search. *Adv. Eng. Softw.* **2016**, *92*, 65–88. [CrossRef]
20. Kaveh, A.; Farhoudi, N. A new optimization method: Dolphin echolocation. *Adv. Eng. Softw.* **2013**, *59*, 53–70. [CrossRef]
21. Hussien, A.G.; Amin, M. A self-adaptive Harris Hawks optimization algorithm with opposition-based learning and chaotic local search strategy for global optimization and feature selection. *Int. J. Mach. Learn. Cyb.* **2022**, *13*, 309–336. [CrossRef]
22. Hussien, A.G.; Oliva, D.; Houssein, E.H.; Juan, A.A.; Yu, X. Binary whale optimization algorithm for dimensionality reduction. *Mathematics* **2020**, *8*, 1821. [CrossRef]
23. Assiri, A.S.; Hussien, A.G.; Amin, M. Ant Lion Optimization: Variants, hybrids, and applications. *IEEE Access* **2020**, *8*, 77746–77764. [CrossRef]
24. Hussien, A.G.; Amin, M.; Wang, M.; Liang, G.; Alsanad, A.; Gumaei, A.; Chen, H. Crow Search Algorithm: Theory, Recent Advances, and Applications. *IEEE Access* **2020**, *8*, 173548–173565. [CrossRef]
25. Hussien, A.G. An enhanced opposition-based Salp Swarm Algorithm for global optimization and engineering problems. *J. Amb. Intel. Hum. Comp.* **2021**, *13*, 129–150. [CrossRef]

26. Hussien, A.G.; Amin, M.; Abd El Aziz, M. A comprehensive review of moth-flame optimisation: Variants, hybrids, and applications. *J. Exp. Theor. Artif. Intell.* **2020**, *32*, 705–725. [CrossRef]
27. Hashim, F.A.; Hussien, A.G. Snake Optimizer: A novel meta-heuristic optimization Algorithm. *Knowl.-Based Syst.* **2022**, *242*, 108320. [CrossRef]
28. Mostafa, R.R.; Hussien, A.G.; Khan, M.A.; Kadry, S.; Hashim, F. Enhanced COOT Optimization Algorithm for Dimensionality Reduction. In Proceedings of the Fifth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU), Riyadh, Saudi Arabia, 28–29 March 2022. [CrossRef]
29. Jain, M.; Singh, V.; Rani, A. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. *Swarm Evol. Comput.* **2019**, *44*, 148–175. [CrossRef]
30. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1992.
31. Juste, K.; Kita, H.; Tanaka, E.; Hasegawa, J. An evolutionary programming solution to the unit commitment problem. *IEEE T. Power Syst.* **1999**, *14*, 1452–1459. [CrossRef]
32. Rocca, P.; Oliveri, G.; Massa, A. Differential evolution as applied to electromagnetics. *IEEE Antennas Propag. Mag.* **2011**, *53*, 38–49. [CrossRef]
33. Passino, K.M. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Contr. Syst. Mag.* **2002**, *22*, 52–67.
34. Moscato, P.; Mendes, A.; Berretta, R. Benchmarking a memetic algorithm for ordering microarray data. *Biosystems* **2007**, *88*, 56–75. [CrossRef] [PubMed]
35. Kirkpatrick, S.; Gelatt, C.D., Jr.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [CrossRef] [PubMed]
36. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inform. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
37. BİRBİL, Ş.İ.; Fang, S.C. An electromagnetism-like mechanism for global optimization. *J. Glob. Optim.* **2003**, *25*, 263–282. [CrossRef]
38. Javidy, B.; Hatamlou, A.; Mirjalili, S. Ions motion algorithm for solving optimization problems. *Appl. Soft Comput.* **2015**, *32*, 72–79. [CrossRef]
39. Abualigah, L.; Abd Elaziz, M.; Hussien, A.G.; Alsalibi, B.; Jalali, S.M.J.; Gandomi, A.H. Lightning search algorithm: A comprehensive survey. *Appl. Intell.* **2021**, *51*, 2353–2376. [CrossRef]
40. Doğan, B.; Ölmez, T. A new metaheuristic for numerical function optimization: Vortex search algorithm. *Inform. Sci.* **2015**, *293*, 125–145. [CrossRef]
41. Rao, R.V.; Savsani, V.J.; Vakharia, D. Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Comput. Aided Des.* **2011**, *43*, 303–315. [CrossRef]
42. Atashpaz-Gargari, E.; Lucas, C. Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In Proceedings of the IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007; pp. 4661–4667.
43. Ramezani, F.; Lotfi, S. Social-based algorithm (SBA). *Appl. Soft Comput.* **2013**, *13*, 2837–2856. [CrossRef]
44. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE T. Evolut. Comput.* **1997**, *1*, 67–82. [CrossRef]
45. Ning, G.Y.; Cao, D.Q. Improved Whale Optimization Algorithm for Solving Constrained Optimization Problems. *Discret. Dyn. Nat. Soc.* **2021**, *2021*, 8832251. [CrossRef]
46. Nautiyal, B.; Prakash, R.; Vimal, V.; Liang, G.; Chen, H. Improved Salp Swarm Algorithm with Mutation Schemes for Solving Global Optimization and Engineering Problems. *Eng. Comput.* **2021**, 1–23. [CrossRef]
47. Tubishat, M.; Idris, N.; Shuib, L.; Abushariah, M.A.M.; Mirjalili, S. Improved Salp Swarm Algorithm Based on Opposition Based Learning and Novel Local Search Algorithm for Feature Selection. *Expert Syst. Appl.* **2020**, *145*, 113122. [CrossRef]
48. Pelusi, D.; Mascella, R.; Tallini, L.; Nayak, J.; Naik, B.; Deng, Y. An Improved Moth-Flame Optimization Algorithm with Hybrid Search Phase. *Knowl. Based Syst.* **2020**, *191*, 105277. [CrossRef]
49. Naruei, I.; Keynia, F. Wild horse optimizer: A new meta-heuristic algorithm for solving engineering optimization problems. *Eng. Comput.* **2021**. [CrossRef]
50. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
51. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* **2015**, *89*, 228–249. [CrossRef]
52. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]
53. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
54. Singh, N.; Kaur, J. Hybridizing sine-cosine algorithm with harmony search strategy for optimization design problems. *Soft Comput.* **2021**, *25*, 11053–11075. [CrossRef]
55. Li, Y.; Zhao, Y.; Liu, J. Dynamic sine cosine algorithm for large-scale global optimization problems. *Expert Syst. Appl.* **2021**, *177*, 114950. [CrossRef]
56. Wang, S.; Sun, K.; Zhang, W.; Jia, H. Multilevel thresholding using a modified ant lion optimizer with opposition-based learning for color image segmentation. *Math. Biosci. Eng.* **2021**, *18*, 3092–3143. [CrossRef]
57. Abdollahzadeh, B.; Gharehchopogh, F.S.; Mirjalili, S. Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems. *Int. J. Intell. Syst.* **2021**, *36*, 5887–5958. [CrossRef]

58. Chen, H.L.; Yang, C.J.; Heidari, A.A.; Zhao, X.H. An efficient double adaptive random spare reinforced whale optimization algorithm. *Expert Syst. Appl.* **2020**, *154*, 113018. [CrossRef]
59. Dong, H.; Xu, Y.L.; Li, X.P.; Yang, Z.L.; Zou, C.H. An improved antlion optimizer with dynamic random walk and dynamic opposite learning. *Knowl. Based Syst.* **2021**, *216*, 106752. [CrossRef]
60. Suganthan, P.N.; Hansen, N.; Liang, J.J.; Deb, K.; Chen, Y.P.; Auger, A.; Tiwari, S. *Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization*; Technical Report, Nanyang Technological University, Singapore and KanGAL; Kanpur Genetic Algorithms Lab.: Kanpur, India, 2005.
61. Zheng, R.; Jia, H.M.; Abualigah, L.; Wang, S.; Wu, D. An improved remora optimization algorithm with autonomous foraging mechanism for global optimization problems. *Math. Biosci. Eng.* **2022**, *19*, 3994–4037. [CrossRef]
62. García, S.; Fernández, A.; Luengo, J.; Herrera, F. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Inf. Sci.* **2010**, *180*, 2044–2064. [CrossRef]
63. Theodorsson-Norheim, E. Friedman and Quade tests: BASIC computer program to perform nonparametric two-way analysis of variance and multiple comparisons on ranks of several related samples. *Comput. Biol. Med.* **1987**, *17*, 85–99. [CrossRef]
64. Coello, C.A.C. Use of a self-adaptive penalty approach for engineering optimization problems. *Comput. Ind.* **2000**, *41*, 113–127. [CrossRef]
65. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.* **2016**, *96*, 120–133. [CrossRef]
66. Mirjalili, S. The ant lion optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [CrossRef]
67. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2016**, *27*, 1053–1073. [CrossRef]
68. Tanabe, R.; Fukunaga, A.S. Improving the search performance of SHADE using linear population size reduction. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 1658–1665.
69. Baykasoğlu, A.; Akpinar, Ş. Weighted superposition attraction (WSA): A swarm intelligence algorithm for optimization problems–part 2: Constrained optimization. *Appl. Soft Comput.* **2015**, *37*, 396–415. [CrossRef]
70. Abualigah, L.; Yousri, D.; Abd Elaziz, M.; Ewees, A.A.; Al-Qaness, M.A.; Gandomi, A.H. Aquila optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [CrossRef]
71. 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]
72. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [CrossRef]
73. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–68. [CrossRef]
74. Ray, T.; Saini, P. Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Eng. Optimiz.* **2001**, *33*, 735–748. [CrossRef]
75. Abualigah, L.; Diabat, A.; Mirjalili, S.; Abd Elaziz, M.; Gandomi, A.H. The arithmetic optimization algorithm. *Comput. Method Appl. M.* **2021**, *376*, 113609. [CrossRef]
76. Gu, L.; Yang, R.; Tho, C.H.; Makowskit, M.; Faruquet, O.; Li, Y.L. Optimisation and robustness for crashworthiness of side impact. *Int. J. Vehicle Des.* **2001**, *26*, 348–360. [CrossRef]
77. Yildiz, B.S.; Pholdee, N.; Bureerat, S.; Yildiz, A.R.; Sait, S.M. Enhanced grasshopper optimization algorithm using elite opposition-based learning for solving real-world engineering problems. *Eng. Comput.* **2021**. [CrossRef]
78. Yang, X.S.; He, X. Firefly algorithm: Recent advances and applications. *Int. J. Swarm Intell. R.* **2013**, *1*, 36–50. [CrossRef]