

Article

Hybrid Reptile Search Algorithm and Remora Optimization Algorithm for Optimization Tasks and Data Clustering

Khaled H. Almotairi ^{1,*} and Laith Abualigah ^{2,3,*} 

¹ Computer Engineering Department, Umm Al-Qura University, Makkah 24382, Saudi Arabia

² Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan

³ School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang 11800, Malaysia

* Correspondence: khmotairi@uqu.edu.sa (K.H.A.); Alighah.2020@gmail.com (L.A.)

Abstract: Data clustering is a complex data mining problem that clusters a massive amount of data objects into a predefined number of clusters; in other words, it finds symmetric and asymmetric objects. Various optimization methods have been used to solve different machine learning problems. They usually suffer from local optimal problems and unbalance between the search mechanisms. This paper proposes a novel hybrid optimization method for solving various optimization problems. The proposed method is called HRSA, which combines the original Reptile Search Algorithm (RSA) and Remora Optimization Algorithm (ROA) and handles these mechanisms' search processes by a novel transition method. The proposed HRSA method aims to avoid the main weaknesses raised by the original methods and find better solutions. The proposed HRSA is tested on solving various complicated optimization problems—twenty-three benchmark test functions and eight data clustering problems. The obtained results illustrate that the proposed HRSA method performs significantly better than the original and comparative state-of-the-art methods. The proposed method overwhelmed all the comparative methods according to the mathematical problems. It obtained promising results in solving the clustering problems. Thus, HRSA has a remarkable efficacy when employed for various clustering problems.



Citation: Almotairi, K.H.; Abualigah, L. Hybrid Reptile Search Algorithm and Remora Optimization Algorithm for Optimization Tasks and Data Clustering. *Symmetry* **2022**, *14*, 458. <https://doi.org/10.3390/sym14030458>

Academic Editor: Theodore E. Simos

Received: 21 January 2022

Accepted: 18 February 2022

Published: 24 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Unsupervised learning methods are instrumental in machine learning because they may explore data without any prior knowledge of them, i.e., there are no labels linked with the data [1]. These algorithms try to represent the data's underlying mechanism or pattern, which may be helpful for things such as decision making and forecasting future inputs. Clustering and feature methods are classic examples of unsupervised algorithms [2,3].

Clustering is an important unsupervised method that uses a number of data objects or items to discover homogenous groupings [4]. Clustering is used to split data items into groups so that objects in the same cluster are similar to one another and different from objects in other clusters. Algorithms for clustering have been utilized in a wide range of applications. They are used in biology to extract interesting patterns from gene expression [5,6]. Additionally, they are used to divide sensor networks into groups in wireless sensor networks and in information retrieval for grouping text content and generating thematic hierarchies. For a product search task, clustering can be utilized to group people or objects [7].

The data and available information are the root of the fast development model of today [8], and are extensively employed in various information technology applications such as manufacturing, marketing, and commerce. Because there are so many data, data mining is one of the most critical methods for extracting meaningful information. Data

mining is a revolutionary idea utilized to tailor knowledge across numerous industries, such as medical records examination, client transaction analysis, and market surveying programs [9]. The clustering and classification algorithms are the two most essential requirements for extracting information from data mining. As a result, data clustering has become a valuable and demanding process for generating clusters where data are clustered together. Data clustering is achieved by using two clustering mechanisms: hierarchical and partitional clustering [10,11]. The data items are grouped hierarchically, either in the shape of a tree or a cluster analysis, in a clustering algorithm. The partitional clustering technique, on the other hand, produces non-overlapping groups. This method has a propensity to cope with noise and outliers, manageability, integration, usability, and other issues in real-world applications. K-means clustering, Fuzzy c-means method, density accurate and robust, and so on are some of the most frequently used classification techniques [12,13].

Several related works have been conducted based on using optimization techniques to solve the clustering problems, which is worth mentioning [14,15]. They motivated us to propose a new efficient method to deal with various data clustering issues.

A novel Class Topper Optimization (CTO) method is presented in [16]. The cognitive intelligence of pupils in a class inspired the optimization method. A population-based search technique is used. The answer narrows towards the optimal solution in this technique. This may lead to a worldwide optimal solution. A clustering challenge is explored to validate the algorithm's performance. For real-time testing, five common data sets are examined. A comparison of the performance of the proposed approach to many well-known existing optimization techniques reveals that it outperforms them all.

A novel clustering approach is presented based on the evolution particle swarm optimization (EPSO) algorithm [17]. The suggested approach is based on the development of swarm generations. At first, the particles are equally dispersed in the input data set, and a new swarm population emerges after a defined number of repetitions. This paper covers the novel technique and its initial implementation and testing using real clustering benchmark data. The findings demonstrate that the technique creates compact clusters and is effective.

A modernized firefly technique is combined with the well-known PSO approach to handle automated data clustering difficulties [18]. The proposed fusion technique is designed against four standard metaheuristic algorithms from the literature using twelve common datasets and the two moons dataset to see how well it performs. The thorough computational tests and results analysis reveals that the suggested approach outperforms the standard published methods.

The ABCWOA method, which uses Random Memory (RM) and Elite Memory (EM) to overcome investigation problems and late convergence in ABC, is presented in [19]. The search phase for the bait in the whale optimization algorithm (WOA) is employed by RM in the ABCWOA method, and EM is also used to boost convergence. The ABCWOA algorithm outperforms other metaheuristic algorithms, according to the results.

This study suggests a novel heuristic strategy based on the big bang–big crunch algorithm for clustering issues [20]. In comparison to previous heuristic strategies, the suggested method not only utilizes the use of the heuristic nature to reduce standard clustering algorithms such as k-means, but it also benefits from the memory-based scheme. Furthermore, the proposed algorithm's system is assessed using various benchmark functions, including well datasets. The experimental findings reveal that the suggested technique outperforms similar algorithms substantially.

This study proposes K–NM–PSO, a hybrid approach that combines the K-means algorithm, Nelder–Mead simplex search, and PSO [21]. The K–NM–PSO method, like the K-means algorithm, looks for cluster centers in any data set. However, it can also discover global optima effectively and rapidly. The results suggest that K–NM–PSO is reliable and appropriate for data clustering. Other optimization methods can be used to solve the data clustering problems. Additionally, many optimization methods have been used to solve various problems [22]. It is also worth mentioning that the optimization methods

proved their ability in solving many other problems such as solve complex engineering optimization problems [23], reactive power planning problems [24], parameter estimation problems [25], voltage-constrained power problems [26], and others.

The Reptile Search Algorithm (RSA) is a recent and novel optimizer that was proposed by Abualigah et al. in [27]. This method mimics the behavior of crocodiles on two principal activities—encircling and hunting—each of which is performed in two main strategies: (1) encircling: high walking or belly walking; (2) hunting: hunting coordination or hunting cooperation. The Remora Optimization Algorithm (ROA) is a recent and novel optimizer that has been proposed by Jia et al. in [28], a new bionics-based optimization method. The main inspiration of the ROA is Remora's behavior.

Various optimization methods have been employed in the literature to solve different machine learning problems, especially clustering problems. The proposed method's primary motivation is that the recent studies' that used optimization methods for solving similar problems yielded performances that were not good enough, and a new, improved method can find new optimal solutions. Optimization methods usually suffer from optimal local problems and unbalance between the search mechanisms (i.e., exploration and exploitation). Moreover, we selected the most reputed methods to incorporate them and created a new hybrid approach to yield new and better results. This paper proposed a novel hybrid optimization search method for solving challenging optimization problems. The proposed method is called HRSA, which combines the original Reptile Search Algorithm (RSA) and Remora Optimization Algorithm (ROA) and handles these mechanisms' search processes by a novel transition method. The proposed HRSA method solves the main weaknesses raised by the original methods and finds better solutions. The proposed HRSA is tested on two classes of optimization problems: twenty-three benchmark test functions and eight real-world data clustering problems. The results illustrate that the proposed HRSA method performs significantly better than the original and comparative state-of-the-art methods. The proposed method overwhelmed all the comparative methods according to the mathematical problems. It obtained promising results in solving the clustering problems. Thus, HRSA has a remarkable ability to be employed for various clustering problems.

The major contributions in this paper are covered in the following points.

- A novel hybrid optimization method is proposed using the original Reptile Search Algorithm (RSA) and Remora Optimization Algorithm (ROA).
- A new transition method is proposed to handle the mechanisms' search processes and help the proposed method in enabling the suitable search operator during the optimization process.
- The performance of the proposed HRSA method is tested on several benchmark functions to show the mai.
- A real-world problem, data clustering, is used to prove further the proposed HRSA's ability to deal with complicated problems, finding symmetric and asymmetric objects.
- The results showed the superiority of the proposed method in solving the given problems compared to other various state-of-the-art methods.

In the rest of the paper, Section 2 presents the background of the used methods and the procedure of the proposed method. Section 3 shows the experiments, results, and discussion. Section 4 gives the research conclusions and potential future work.

2. Background

This section presents the original methods used in the proposed method. Additionally, the main procedure of the proposed hybrid Reptile Search Algorithm (RSA) and Remora Optimization Algorithm (ROA) with a novel transition mechanism (HRSA) is presented.

2.1. Reptile Search Algorithm (RSA)

This section presents the original Reptile Search Algorithm (RSA) and its procedure. The basic Reptile Search Algorithm (RSA) is described in its exploration (global search) and

exploitation (local search) stages, which were inspired by the encircling mechanics, hunting processes, and social behavior of crocodiles in real life [27].

2.1.1. Encircling Phase (Exploration)

The exploratory behavior (encircling) of the RSA is introduced in this section. Crocodiles engage in two movements when encircling: high walking and belly walking, according to their encircling behavior [29].

The RSA switches between exploration and exploitation search phases based on four scenarios: splitting the number of iterations into four parts; dividing the total number of iterations into four parts. The RSA exploration mechanisms investigate the search regions and approach to discover a better answer based on two major search techniques.

One condition must be met throughout this phase of the search. The high walking search method is performed according to $t \leq \frac{T}{4}$, and the belly walking search method is performed according to $t \leq 2\frac{T}{4}$ and $t > \frac{T}{4}$. The position-updating process is presented in Equation (1).

$$x_{(i,j)}(t+1) = \begin{cases} Best_j(t) \times -\eta_{(i,j)}(t) \times \beta - R_{(i,j)}(t) \times rand, & t \leq \frac{T}{4} \\ Best_j(t) \times x_{(r_1,j)} \times ES(t) \times rand, & t \leq 2\frac{T}{4} \text{ and } t > \frac{T}{4} \end{cases} \quad (1)$$

where $Best_j(t)$ is the best-obtained solution, $rand$ is a random number, t is the current iteration, and T is the maximum iterations. $\eta_{(i,j)}$ is the hunting parameter determined by Equation (2). β is a parameter fixed to 0.1. The reduce function ($R_{(i,j)}$) is determined by Equation (3). $r_1 - r_4$ are random numbers, $x_{(r_1,j)}$ is a random position, and N is the used solutions. Evolutionary Sense ($ES(t)$) is a probability parameter determined by Equation (4).

$$\eta_{(i,j)} = Best_j(t) \times P_{(i,j)}, \quad (2)$$

$$R_{(i,j)} = \frac{Best_j(t) - x_{(r_2,j)}}{Best_j(t) + \epsilon}, \quad (3)$$

$$ES(t) = 2 \times r_3 \times \left(1 - \frac{1}{T}\right), \quad (4)$$

where ϵ a small value. $P_{(i,j)}$ is a difference parameter determined by Equation (5).

$$P_{(i,j)} = \alpha + \frac{x_{(i,j)} - M(x_i)}{Best_j(t) \times (UB_{(j)} - LB_{(j)}) + \epsilon}, \quad (5)$$

where $M(x_i)$ represents the average positions determined by Equation (6). $UB_{(j)}$ and $LB_{(j)}$ are the upper and lower boundaries. α is a parameter fixed to 0.1.

$$M(x_i) = \frac{1}{n} \sum_{j=1}^n x_{(i,j)}, \quad (6)$$

2.1.2. Hunting Phase (Exploitation)

The exploitative behavior of RSA is discussed in this section. Crocodiles use two hunting techniques: hunting coordination and hunting collaboration, according to their hunting behavior.

The searching in this phase (hunting coordination) is executed and determined according to $t \leq 3\frac{T}{4}$ and $t > 2\frac{T}{4}$; otherwise, the hunting cooperation is executed according to $t \leq T$ and $t > 3\frac{T}{4}$. The position-updating processes are presented in Equation (7):

$$x_{(i,j)}(t+1) = \begin{cases} Best_j(t) \times P_{(i,j)}(t) \times rand, & t \leq 3\frac{T}{4} \text{ and } t > 2\frac{T}{4} \\ Best_j(t) - \eta_{(i,j)}(t) \times \epsilon - R_{(i,j)}(t) \times rand, & t \leq T \text{ and } t > 3\frac{T}{4} \end{cases} \quad (7)$$

where $Best_j(t)$ is the best obtained solution, and $\eta_{(i,j)}$ is the hunting parameter determined by Equation (2). $P_{(i,j)}$ is a difference parameter determined by Equation (5). $\eta_{(i,j)}$ is the hunting parameter determined by Equation (2). $R_{(i,j)}$ is determined by Equation (3).

2.2. Remora Optimization Algorithm (ROA)

This section presents the original Remora Optimization Algorithm (ROA) and its procedure, which is as follows [28].

2.2.1. Free Travel (Exploration)

SFO Strategy

The formulation for this algorithm's location update was modeled based on the algorithm's elite notion, yielded by Equation (8).

$$R_i^{t+1} = R_{best}^t - \left(rand \times \left(\frac{R_{best}^t - R_{rand}^t}{2} \right) - R_{rand}^t \right) \quad (8)$$

where R_{rand}^t is a random location.

Experience Attack

To evaluate whether or not it is required to replace the host, the tuyu must regularly take modest steps around the host, analogous to the development of knowledge. The formula for modeling the principles as mentioned above is as follows:

$$R_{att} = R_i^t - (R_i^t - R_{pre}) \times randn \quad (9)$$

where R_{pre} is the position of the previous iteration, and R_{att} is a tentative step.

The evaluation of the fitness function of the present solution $f(R_i^t)$ and the attempted solution $f(R_{att})$ is described as the decision of this step. As an example, when addressing the minimal problem, if the fitness function value produced by the proposed solution is less than the present solution,

$$f(R_i^t) > f(R_{att}) \quad (10)$$

Remora chooses a different mechanism for local optima, as shown in the following section. It returns to host choosing if the fitness function value of the attempted solution is greater than that of the existing solution.

$$f(R_i^t) < f(R_{att}) \quad (11)$$

2.2.2. Eat Thoughtfully (Exploitation)

WOA Strategy

The location update formula of Remora connected to the whale was recovered using the original WOA technique, as seen in the Equations below:

$$R_{i+1} = D \times e^\alpha \times \cos(2\pi\alpha) + R_i \quad (12)$$

$$\alpha = rand \times (a - 1) + 1 \quad (13)$$

$$a = -(1 + \frac{t}{T}) \quad (14)$$

$$D = |R_{best} - R_i| \quad (15)$$

Their locations might be considered the same when a Remora is on a whale in the broader solution space. D is the space between both the hunter and the prey, α is a random value in the range $[-1,1]$, and a is a number that decreases exponentially between $[-2, -1]$.

Host Feeding

The exploitation procedure is further subdivided into host feeding. At this point, the optimal solution can be condensed to the host's location area. Incremental steps can be thought of as moving on or around the host, which can be mathematically characterized as:

$$R_i^t = R_i^t + A \quad (16)$$

$$A = B \times (R_i^t - C \times R_{best}) \quad (17)$$

$$B = 2 \times V \times rand - V \quad (18)$$

$$V = 2 \times (1 - \frac{t}{T}) \quad (19)$$

A was employed here to represent a tiny movement connected to the size space of the host and remora. A Remora factor (C) was utilized to limit the position of Remora to distinguish between the host and Remora. If the size of the host is 1, then the volume of the Remora is about a percent of the host's volume.

2.3. The Proposed HRSA Method

This section presents the main procedure of the proposed hybrid Reptile Search Algorithm (RSA) and Remora Optimization Algorithm (ROA) with a novel transition mechanism (HRSA).

The suggested HRSA is used to solve various issues using two major search strategies and a new mean transition mechanism. The traditional RSA is the principal search technique, which has strong global search capabilities but occasionally suffers from local search problems, premature convergence, and disequilibrium between global and local search methods. As a result, the second search strategy, the ROA, avoids the local search issue and premature convergence. The strategy can improve the RSA's searchability by generating new local solutions based on the best solutions currently available. Furthermore, a new mean transition mechanism is provided to govern the execution of the search methods (i.e., RSA and ROA) in the proposed HRSA to tackle the disequilibrium problem between the global and local search techniques. As a result, the search space may be efficiently expanded by incorporating new approaches from other places. These solutions inspire the proposed HRSA to use more robust solutions to obtain better results.

2.3.1. Initialization Phase

The optimization process in RSA begins with a set of candidate solutions (X) created stochastically, as indicated in Equation (20). The best obtained solution is deemed nearly optimal in each iteration.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & x_{2,j} & \cdots & x_{2,n} \\ \cdots & \cdots & x_{i,j} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & x_{N-1,j} & \cdots & x_{N-1,n} \\ x_{N,1} & \cdots & x_{N,j} & x_{N,n-1} & x_{N,n} \end{bmatrix} \quad (20)$$

where X is a collection of the solutions that are created by using Equation (21), $x_{i,j}$ is the j^{th} position of the i^{th} solution, N is the number of solutions, and n is the dimension size.

$$x_{ij} = rand \times (UB - LB) + LB, j = 1, 2, \dots, n \quad (21)$$

where $rand$ is a random value and LB and UB denote the lower and upper bound, respectively.

2.3.2. The Proposed Mean Transition Mechanism (MTM)

The mean transition mechanism (MTM) is devised and given in Algorithm 1 in this section. This technique is used to control the search process and the transition between the RSA and the MT. The transition from one search process to the next is quite delicate. It necessitates an effective way to change the update procedures across the various approaches. The suggested MTM's fundamental concept is to regulate the search techniques when the fitness function does not increase after five iterations (I). When no gains can be made by experimentation, the number of iterations changes.

Algorithm 1 The proposed Mean Transition Mechanism (MTM).

```

1: Initialize the TM parameter value (TM = 0).
2: sumFF = 0;
3: for ( $t = 1$  to  $T$ ) do
4:   sumFF = sumFF + currentFF
5:   C = C + 1;
6:   if (currentFF  $\neq$  sumFF) then
7:     if (C > I) then
8:       TM = flip(TM);
9:       sumFF = 0;
10:      C = 0;
11:    end if
12:   end if
13: end for
```

TM is a binary variable utilized to adjust the search process between the RSA and the MT, $sumFF$ is a variable used to calculate the mean fitness function values, $currentFF$ is the current fitness function value, C is a counter, I is the number of iterations to change when no improvements and it can be adjusted by experiment, and $flip$ is a function to flip the TM value from 0 to 1 or vice versa, as described in Algorithm 1.

2.3.3. The Detailed Process of the HRSA

This section goes through the proposed technique in detail. The main purpose of the proposed strategy is to obtain better results than those yielded by the existing approaches. We also want to avoid the defects presented by the original processes, such as local search issues, convergence constraints, and search balance concerns.

To conclude, the proposed HRSA starts with the creation of a random set of solutions. During the renewal process, the HRSA's search criteria explore possible placements of the current best solution. Each answer progresses to the next stage of the procedure. According to Figure 1, the suggested HRSA uses the Reptile Search Algorithm (RSA) and the Remora Optimization Algorithm (ROA) approach. Each iteration will update and improve the potential solutions using one of these search methodologies.

The suggested HRSA's search techniques are divided into two categories, as shown Figure 1: RSA and ROA. Following this, the RSA's search techniques are divided into global and local search methods. There are two search techniques for each method: (1) at the global level, with high and walking tactics; and (2) hunting coordination and cooperation at the local level. Candidate solutions try to figure out what is going on in the search space $t \leq \frac{T}{2}$ and seek to discover the near-optimal solution if $t > \frac{T}{2}$. For the first section, if $TM == 0$, the search process of the RSA will be performed; otherwise, the search process of the ROA will be conducted. In the exploration step of the RSA, the first search strategy from the global search methods is performed when $t \leq \frac{T}{4}$, and the second strategy from the global search processes are accomplished when $t \leq \frac{3T}{4}$ and $t > \frac{T}{4}$. In the exploitation of the conventional RSA, the first search approach from the local methods is carried out when $t \leq \frac{3T}{4}$ and $t > \frac{T}{4}$; otherwise, the second search approach from the local methods is

executed, when $t \leq T$ and $t > 3\frac{T}{4}$. Finally, the HRSA is finished when it reaches the end criterion. The complete rule of the recommended HRSA is shown in Figure 1.

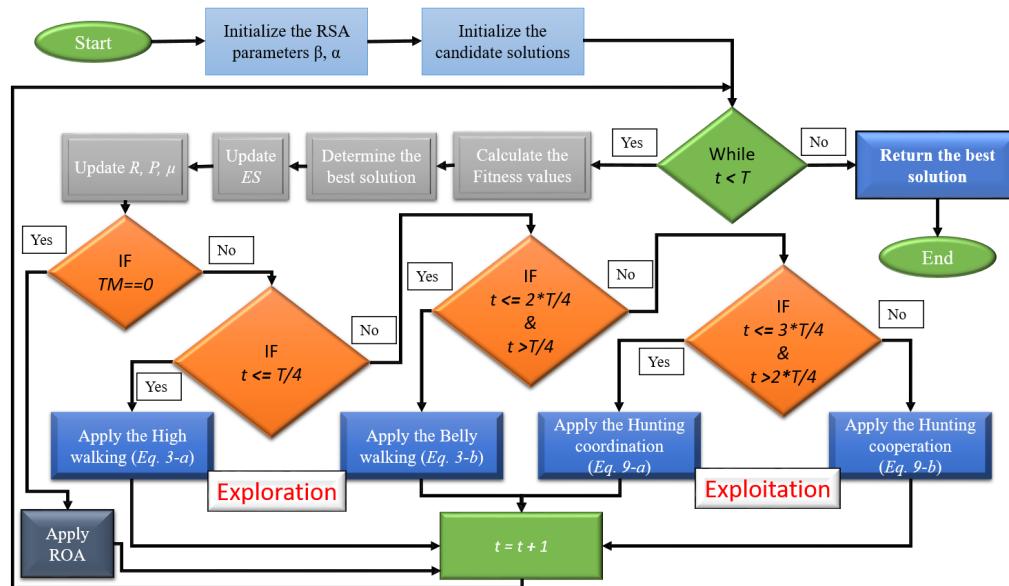


Figure 1. Flowchart of the proposed HRSA.

3. Results and Discussion

In this section, the proposed HRSA method is evaluated using twenty-three common benchmark functions.

3.1. Experimental Settings

The results of the proposed HRSA method are compared with other state-of-the-art methods including Aquila Optimizer (AO), Dwarf Mongoose Optimization Algorithm (DMOA), Whale Optimization Algorithm (WOA), Sine Cosine Algorithm (SCA), Dragonfly Algorithm (DA), Grey Wolf Optimizer (GWO), Particle Swarm Optimizer (PSO), Ant Lion Optimizer (ALO), Reptile Search Algorithm (RSA), Remora Optimization Algorithm (ROA), Arithmetic Optimization Algorithm (AOA), and the proposed HRSA method. All the tested methods are tuned according to the original paper and its parameters, as shown in Table 1. Each algorithm is executed 30 times using 50 solutions and 1000 iterations.

Table 1. Parameter values of the tested algorithms.

No.	Algorithm	Parameter	Value
1	AO	α	0.1
2	DMOA	δ	0.1
3	WOA	-	-
4	SCA	α	Decreased from 2 to 0
5	DA	w	0.05
		s, a , and c	0.2–0.9
		f and e	0.1
6	GWO	Convergence parameter (a)	1
7	PSO	Topology	Linear reduction from 2 to 0
		Cognitive and social constant	Fully connected
		Inertia weight	(C1, C2) 2, 2
		Velocity limit	Linear reduction from 0.9 to 0.1
8	ALO	I ratio	10 ^w
9	RSA	w	2–6
		α	0.1
		β	0.1
10	ROA	α	5
11	AOA	μ	0.5

Twenty-three benchmark problems are used with different characterizations, as shown in Table 2. These test functions are categorized into three main categories, unimodal, multimodal, and multimodal, with fixed dimension functions. They are always used in the domain of machine learning and optimization algorithms [30,31].

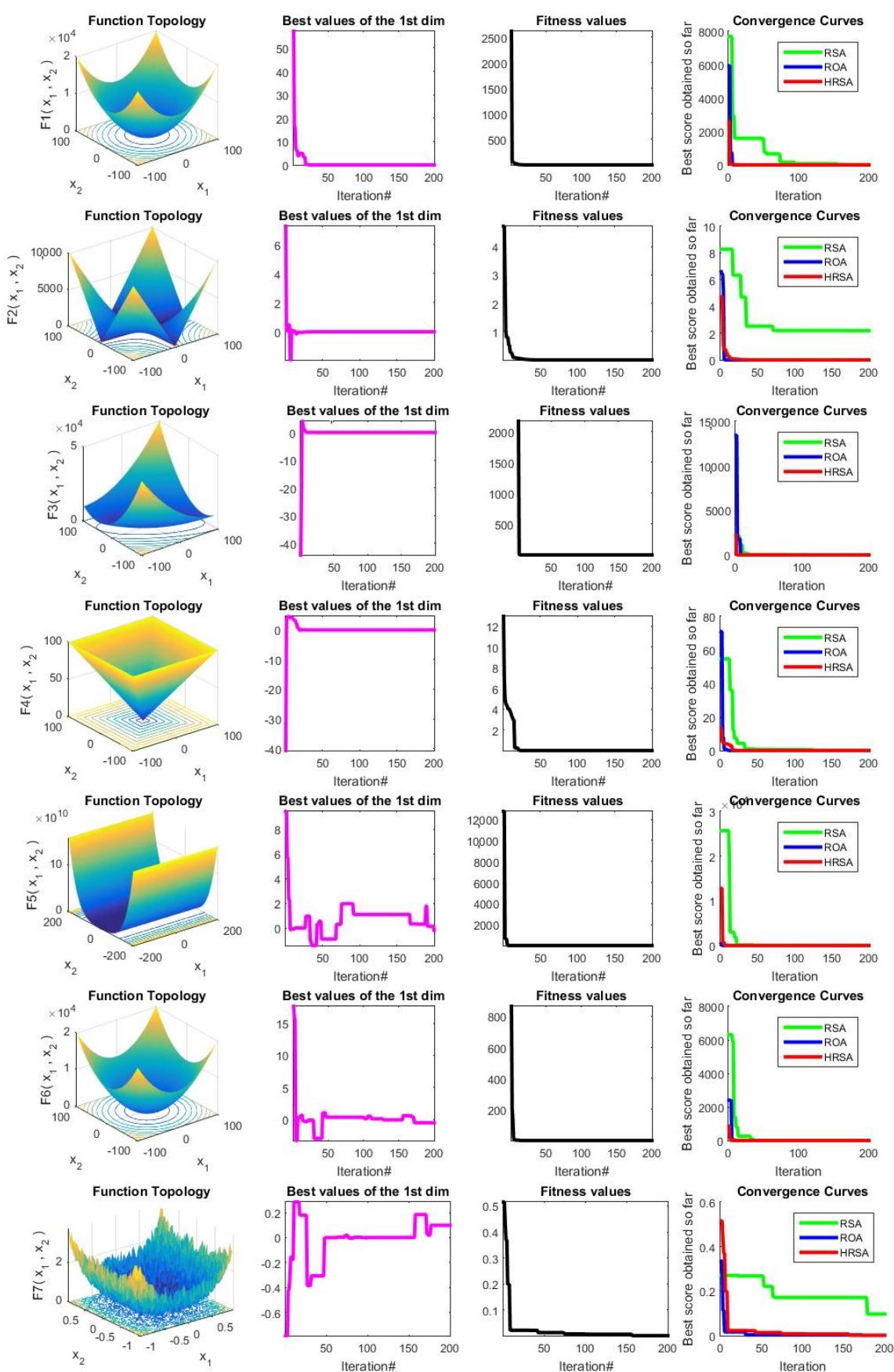
Table 2. Details of the tested benchmark functions.

Function	Description	Dimensions	Range	f_{min}
F1	$f(x) = \sum_{i=1}^n x_i^2$	30	[−100, 100]	0
F2	$f(x) = \sum_{i=0}^n x_i + \prod_{i=0}^n x_i $	30	[−10, 10]	0
F3	$f(x) = \sum_{i=1}^d (\sum_{j=1}^i x_j)^2$	30	[−100, 100]	0
F4	$f(x) = \max_i\{ x_i , 1 \leq i \leq n\}$	30	[−100, 100]	0
F5	$f(x) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2]$	30	[−30, 30]	0
F6	$f(x) = \sum_{i=1}^n [(x_i + 0.5)]^2$	30	[−100, 100]	0
F7	$f(x) = \sum_{i=0}^n ix_i^4 + \text{random}[0, 1)$	30	[−128, 128]	0
F8	$f(x) = \sum_{i=1}^n (-x_i \sin(\sqrt{ x_i }))$	30	[−500, 500]	−418.9829 × n
F9	$f(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[−5.12, 5.12]	0
F10	$f(x) = -20\exp(-0.2\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	10, 50, 100, 500	[−32, 32]	0
F11	$f(x) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}})$ $f(x) = \frac{\pi}{n} \{10\sin(\pi y_1)\} + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1}) + \sum_{i=1}^n u(x_i, 10, 100, 4)]$, where $y_i = 1 + \frac{x_i + 1}{4}, u(x_i, a, k, m) \begin{cases} K(x_i - a)^m & \text{if } x_i > a \\ 0 & -a \leq x_i \geq a \\ K(-x_i - a)^m & -a \leq x_i \end{cases}$	10, 50, 100, 500	[−600, 600]	0
F12	$f(x) = 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)] + \sum_{i=1}^n u(x_i, 5, 100, 4))$	10, 50, 100, 500	[−50, 50]	0
F13	$f(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2} (x_i - a_{ij}) \right)^{-1}$	2	[−65, 65]	1
F14	$f(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[−5, 5]	0.00030
F15	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5, 5]	−1.0316
F16	$f(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	[−5, 5]	0.398
F17	$f(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[−2, 2]	3
F18	$f(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2)$	3	[−1, 2]	−3.86
F19	$f(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2)$	6	[0, 1]	−0.32
F20	$f(x) = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 1]	−10.1532
F21	$f(x) = -\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 1]	−10.4028
F22	$f(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 1]	−10.5363

3.2. Benchmark Function: Experiments

In this section, the results of the comparative methods are presented in several formats to validate their performances. Figure 2 shows the qualitative analysis of the first thirteen benchmark functions (F1–F13) in terms of function topology, trajectory curves of the first dimension values, the average fitness function values, and the convergence curves of the original methods (i.e., RSA and ROA) and the proposed HRSA method.

It is clear from Figure 2 that the proposed method obtained promising results, and it can be an excellent alternative in this domain. The proposed method reaches near-optimal solutions for all the tested functions according to the functions typologies. Additionally, the trajectory curves of the first dimension values show that the proposed method acceptably changes the values. This refers to the diversity of the position-updating mechanisms. The proposed method obtained the minimal values regarding the average fitness function values. The convergence is always near the optimal area. Finally, the convergence curves show that the proposed method overwhelmed the original methods in all the tested problems.

**Figure 2. Cont.**

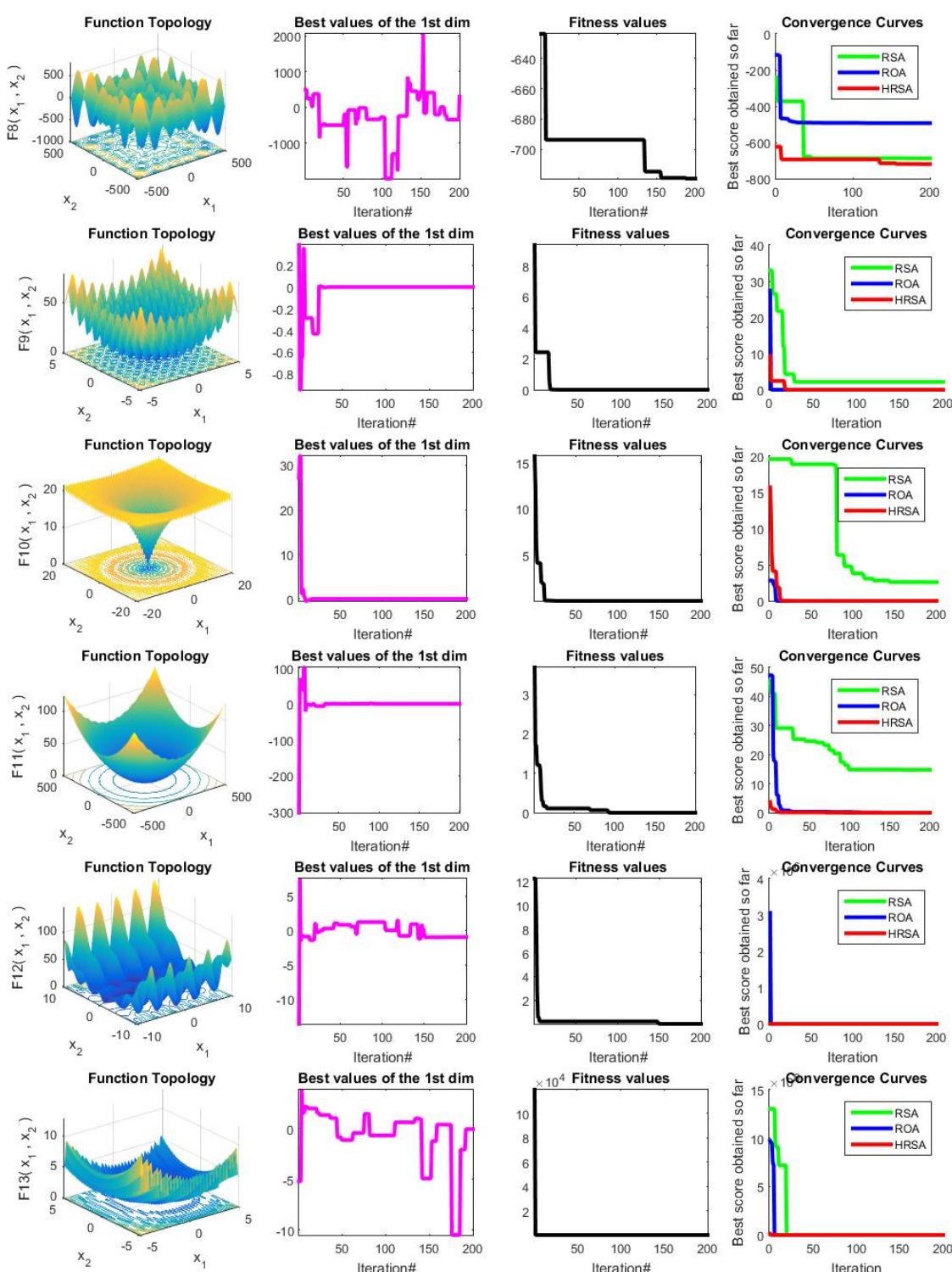


Figure 2. Qualitative analysis of F1–F13.

Table 3 shows the proposed HRSA method in solving thirteen benchmark functions with different population sizes. This experiment was conducted to find the optimal number of solutions to proceed with the following experiments. It is clear from Table 3 that the proposed HRSA obtained the best results when the population size was equal to 50, followed by a population of 45. According to the given results from the Friedman ranking test, when the population size is equal to 50, the proposed HRSA is ranked as the first best method followed by the number of solutions 45, 35, 40, 30, 25, 20, 15, 10, and 5. Thus, the proposed method obtains better results when the number of solutions is larger.

Table 3. The results of several population sizes on F1–F13.

Table 3. Cont.

Function	Measure	5	10	15	20	25	30	35	40	45	50
F12	Worst	4.0854E-02	2.2289E-04	1.3389E-03	6.8061E-04	5.6103E-04	1.7179E-04	1.0947E-04	7.3013E-04	5.0295E-05	2.0846E-05
	Average	1.7935E-02	6.9673E-05	6.5105E-04	2.9193E-04	2.0389E-04	6.2899E-05	4.1642E-05	2.6116E-04	3.0538E-05	5.6652E-06
	Best	4.3072E-03	7.2701E-06	2.2544E-06	1.9810E-05	7.6314E-07	3.2039E-06	3.2509E-07	5.7498E-06	7.3791E-06	2.7871E-07
	STD	1.7213E-02	1.0248E-04	7.0379E-04	3.0985E-04	2.5034E-04	7.5121E-05	4.7190E-05	3.4009E-04	1.7759E-05	1.0124E-05
	p-value	8.2358E-02	2.6020E-01	1.1637E-01	1.1430E-01	1.6466E-01	1.8175E-01	1.8660E-01	1.8382E-01	5.0921E-02	1.0000E+00
	h	0	0	0	0	0	0	0	0	0	0
	Rank	10	5	9	8	6	4	3	7	2	1
	Mean ranking	7.6154E+00	6.8462E+00	5.7692E+00	5.2308E+00	5.2308E+00	3.9231E+00	2.7692E+00	2.8462E+00	2.5385E+00	1.8462E+00
F13	Final ranking	10	9	8	6	6	5	3	4	2	1
	Worst	7.4368E-02	1.5589E-02	7.8932E-04	2.0129E-03	2.2940E-03	2.1774E-04	2.3244E-04	2.0152E-04	8.1074E-04	1.0923E-04
	Average	2.3931E-02	3.9380E-03	2.2898E-04	8.3051E-04	8.9919E-04	1.3918E-04	8.5703E-05	1.1051E-04	3.0715E-04	7.3491E-05
	Best	2.4372E-03	4.2232E-06	2.0465E-05	1.4455E-04	3.8590E-05	6.9042E-05	9.0061E-06	3.9783E-05	1.9733E-05	5.4162E-07
	STD	3.4057E-02	7.7679E-03	3.7411E-04	8.8825E-04	1.0825E-03	6.1273E-05	1.0189E-04	7.0518E-05	3.4733E-04	5.0235E-05
	p-value	2.1075E-01	3.5817E-01	4.4154E-01	1.3969E-01	1.7838E-01	1.4839E-01	8.3690E-01	4.2532E-01	2.3136E-01	NaN
	h	0	0	0	0	0	0	0	0	0	NaN
	Rank	10	9	5	7	8	4	2	3	6	1

Table 4 shows the results of the comparative methods and the proposed HRSA for solving thirteen benchmark functions problems (F1–F13) when the dimension size is equal to 10. It is clear from this table that the proposed method obtained more promising results compared to the other comparative methods. This outcome proved the ability of the modified method to avoid the main problems of the original methods. According to the Wilcoxon signed-rank test, the proposed HRSA overwhelmed almost all the comparative methods in all the given problems. For example, the proposed HRSA significantly overwhelmed DOMA and DA in F1. Additionally, it significantly improved compared to ROA, PSO, GWO, DA, DMOA, and WOA in F4. According to the Friedman ranking test, the proposed HRSA is ranked as the first-best method, followed by the AO method as the second-best method, the RSA as the third-best method, the GWO as the fourth-best method, the AOA as the fifth-best method, the WOA as the sixth-best method, the PSO as the seventh-best method, the ROA as the eighth-best method, the DMOA as the ninth-best method, the SCA as the tenth-best method, the ALO as the eleventh-best method, and finally, the DA as the twentieth-best method. We concluded that the proposed method can find better solutions than the other methods' solutions.

Table 5 shows the results of the comparative methods and the proposed HRSA for solving thirteen benchmark functions problems (F1–F13) when the dimension size is equal to 100. It is clear from this table that the proposed method achieved better results compared to the other comparative methods for these cases. This result demonstrated the improved method's capacity to overcome the original method's major flaws. According to the Wilcoxon signed-rank test, the proposed HRSA overwhelmed almost all the comparative methods in all the given cases. For example, the proposed HRSA significantly overwhelmed ROA, PSO, GWO, SCA, and DMOA in F1. Additionally, it led to significant improvements compared to AOA, ROA, RSA, ALO, PSO, GWO, SCA, DMOA, and WOA in F10. According to the Friedman ranking test, the proposed HRSA is ranked as the first-best method, followed by the AO method as the second-best method, the RSA as the third-best method, the WOA as the fourth-best method, the GWO as the fifth-best method, the AOA as the sixth-best method, the PSO as the seventh-best method, the DMOA as the eighth-best method, the ROA as the ninth-best method, the ALO as the tenth-best method, the DA as the eleventh-best method, and finally, the SCA as the twentieth-best method. We came to the conclusion that the proposed HRSA can find better solutions than previous methods when the dimensions are greater (i.e., Dim = 100).

Table 4. The results of the comparative methods and the proposed HRSA methods for F1–F13, Dim = 10.

Fun	Measure	Comparative Algorithms											
		AO	DMOA	WOA	SCA	DA	GWO	PSO	ALO	RSA	ROA	AOA	HRSA
F1	Worst	1.1571E-100	2.1046E-09	7.6673E-63	2.6611E-09	2.6788E+01	1.1340E-47	2.2231E-17	2.2742E-07	9.5757E-56	6.8209E-11	5.2731E-70	2.0773E-108
Average		2.8927E-101	1.1143E-09	1.9168E-63	7.2690E-10	1.7523E+01	3.2797E-48	1.0342E-17	9.2064E-08	2.9793E-56	2.3922E-11	1.3183E-70	5.1931E-109
Best		3.0073E-116	5.0980E-10	1.1259E-72	4.4844E-13	8.6835E+00	3.8585E-51	5.8258E-20	1.8748E-08	6.6334E-60	1.9429E-12	3.2582E-152	3.4580E-138
STD		5.7855E-101	6.8881E-10	3.8336E-63	1.2935E-09	8.6400E+00	5.3931E-48	1.1728E-17	9.3965E-08	4.5319E-56	3.0232E-11	2.6365E-70	1.0386E-108
p-value		3.5592E-01	1.7790E-02	3.5592E-01	3.0400E-01	6.6787E-03	2.6956E-01	1.2823E-01	9.7758E-02	2.3659E-01	1.6461E-01	3.5592E-01	NaN
h	0	1	0	0	1	0	0	0	0	0	0	0	NaN
Rank	2	10	4	9	12	6	7	11	5	8	3	1	1
F2	Worst	6.3194E-60	1.6949E-02	7.4584E-52	3.5524E-09	2.1903E+00	1.6706E-27	3.9759E-10	5.6065E-01	1.4412E-32	9.6848E-09	0.0000E+00	5.3364E-49
Average		1.5799E-60	4.2476E-03	3.3970E-52	1.3483E-09	1.4024E+00	6.0495E-28	2.3544E-10	1.4063E-01	6.9571E-33	6.0154E-09	0.0000E+00	1.4410E-49
Best		7.3417E-75	1.1089E-05	5.7800E-54	1.5327E-11	2.4793E-01	1.6624E-28	4.9298E-11	1.5065E-05	2.1170E-33	1.5222E-09	0.0000E+00	1.5588E-55
STD		3.1597E-60	8.4673E-03	3.7501E-52	1.6440E-09	8.3182E-01	7.1418E-28	1.4544E-20	2.8002E-01	5.7688E-33	3.4094E-09	0.0000E+00	2.6047E-49
p-value		3.1092E-01	3.5445E-01	3.1197E-01	1.5207E-01	1.5007E-02	1.4118E-01	1.7740E-02	3.5393E-01	5.2432E-02	1.2387E-02	3.1092E-01	NaN
h	0	0	0	0	1	0	1	0	0	1	0	NaN	
Rank	2	10	3	8	12	6	7	11	5	9	1	4	
F3	Worst	2.5481E-104	8.6720E-02	2.6247E+03	7.8583E-02	5.3597E+03	4.1665E-22	6.3364E-05	3.6192E-01	5.3170E-25	3.6758E-02	5.8269E-68	5.7223E-116
Average		6.3723E-105	2.1726E-02	9.6402E-02	3.7630E-02	2.6504E-03	2.2794E-22	1.9621E-05	2.3056E-01	1.3293E-25	1.4828E-02	1.4571E-68	1.4306E-116
Best		7.0353E-112	1.4915E-05	1.5954E-05	3.1383E-06	2.5278E-01	1.3713E-23	1.8828E-06	8.6392E-02	2.7175E-32	9.8927E-04	6.3905E-108	1.7975E-141
STD		1.2739E-104	4.3330E-02	1.1759E-03	4.2007E-02	3.0089E-03	1.7380E-22	2.9324E-05	1.4879E-01	2.6585E-25	1.5408E-02	2.9132E-68	2.8611E-116
p-value		3.5574E-01	3.5467E-01	1.5221E-01	1.2338E-01	1.2859E-01	3.9423E-02	2.2931E-01	3.5592E-01	1.0260E-01	3.5578E-01	NaN	NaN
h	0	0	0	0	1	0	1	0	0	0	0	NaN	
Rank	2	8	11	9	12	5	6	10	4	7	3	1	
F4	Worst	4.3892E-61	2.9025E-05	1.6160E+01	6.5799E-03	5.9530E+00	2.8186E-15	7.4656E-04	9.9430E-02	5.3933E-20	4.2407E+00	8.1594E-18	3.0089E-51
Average		1.0973E-61	2.3631E-05	9.1448E+00	3.2700E-03	4.1841E+00	1.9128E-15	3.9903E-04	3.7199E-02	2.5279E-20	2.7561E+00	2.0398E-18	7.7464E-52
Best		9.6316E-79	1.6075E-05	1.3178E-01	2.3195E-04	2.7084E+00	9.4618E-16	7.2129E-04	2.7512E-03	1.3331E-21	1.4819E+00	2.2244E-67	6.9049E-60
STD		2.1946E-61	6.2330E-06	6.6791E-00	3.1005E-03	1.6482E+00	1.0353E-15	3.1951E-04	4.3576E-02	2.7085E-20	1.1532E+00	4.0797E-18	1.4890E-51
p-value		3.3717E-01	2.7356E-04	3.3811E-02	7.9449E-02	2.2717E-03	1.0148E-02	4.6670E-02	1.3863E-01	1.1120E-01	3.0628E-03	3.5592E-01	NaN
h	0	1	1	0	1	1	0	0	1	0	0	NaN	
Rank	1	6	12	8	11	5	7	9	3	10	4	2	
F5	Worst	1.0090E-02	6.8474E+02	7.4820E+00	8.7223E+00	7.1798E+02	7.1956E+00	5.8258E+00	3.7504E+02	5.8100E+00	1.1307E+01	8.6967E+00	1.5772E+00
Average		2.9026E-03	1.9907E+02	7.1510E+00	8.1759E+00	2.6870E+02	6.6393E+00	5.2538E+00	1.8256E+02	5.5321E+00	5.9579E+00	8.4321E+00	8.0915E-01
Best		2.3533E-04	8.2388E+00	6.7911E+00	7.8712E+00	7.2106E+01	6.1326E+00	4.5532E+00	5.7113E+00	5.2342E+00	2.0689E+00	8.1501E+00	8.5112E-02
STD		4.7968E-03	3.2628E+02	2.8261E-01	3.7457E-01	3.0211E+02	5.3406E-01	5.3658E-01	2.0333E+02	3.3553E-01	3.9620E+00	2.2871E-01	6.8245E-01
p-value		5.6073E-02	2.6991E-01	2.4973E-06	1.4062E-06	1.2652E-01	1.0441E-05	5.0597E-05	1.2404E-01	1.2291E-05	4.2831E-02	7.2168E-07	NaN
h	0	0	1	1	0	1	1	0	1	1	1	NaN	
Rank	1	11	7	8	12	6	3	10	4	5	9	2	
F6	Worst	3.0958E-04	2.0121E-09	2.2025E-01	7.9699E-01	1.0822E+01	2.5028E-01	1.5409E-02	8.8196E-08	1.3446E-12	2.1400E-10	6.1615E-01	3.4780E-17
Average		1.2661E-04	1.3904E-09	6.2350E-02	5.9809E-01	5.2641E+00	6.2573E-02	4.5459E-03	3.1304E-08	3.5418E-13	6.8957E-11	4.1272E-01	1.3592E-17
Best		1.8530E-06	8.1527E-01	1.8655E-03	4.7048E-01	1.1840E-01	1.3983E-06	1.3847E-04	8.7919E-09	1.8773E-15	1.4982E-12	1.9619E-01	5.6694E-20
STD		1.5073E-04	5.8512E-10	1.0587E-01	1.4410E-01	5.3834E+00	1.2514E-01	7.2704E-03	3.8019E-08	6.6091E-13	9.8755E-11	1.7518E-01	1.4849E-17
p-value		2.6986E-01	2.5767E-01	3.1779E-01	1.7408E-04	9.8511E-02	3.9025E-01	2.5767E-01	2.5767E-01	2.5767E-01	2.5767E-01	3.4818E-03	NaN
h	0	0	0	1	0	0	0	0	0	0	0	NaN	
Rank	6	4	8	11	12	9	7	5	3	2	10	1	
F7	Worst	5.8742E-03	1.7491E-02	3.2814E-03	2.8844E-03	1.3477E-01	5.4463E-04	8.4784E-03	1.8516E-01	7.7024E-04	4.6724E-02	9.2704E-04	2.8372E-04
Average		3.4183E-03	1.0077E-02	2.0021E-03	1.6652E-03	6.3525E-02	4.1147E-04	6.1809E-03	1.0811E-01	4.7066E-04	2.6367E-02	4.7710E-04	1.3225E-04
Best		5.0335E-04	1.3412E-03	4.7093E-04	6.0223E-04	6.4245E-03	2.8956E-04	4.3349E-03	7.1808E-02	3.4456E-04	1.3936E-02	1.3284E-04	1.1502E-05
STD		2.2207E-03	6.9666E-03	1.4820E-03	9.9504E-04	5.9631E-02	1.3949E-04	1.8209E-03	5.1922E-02	2.0081E-04	1.4979E-02	3.6519E-04	1.1275E-04
p-value		2.5427E-02	1.1841E-01	3.2958E-01	1.9972E-01	9.0584E-02	3.5451E-02	1.0270E-01	6.8887E-03	3.8341E-02	2.3070E-02	3.9919E-02	NaN
h	1	0	0	0	0	1	0	1	1	1	1	NaN	
Rank	7	9	6	8	11	2	7	5	3	2	10	1	
F8	Worst	-1.9843E+03	-2.4441E+03	-2.2903E+03	-1.8464E+03	-2.6001E+03	-1.9888E+03	-1.8626E+03	-1.9257E+03	-2.7610E+03	-2.5286E+03	-2.1284E+03	-1.7313E+03
Average		-2.3457E+03	-2.7747E+03	-2.8844E+03	-1.9811E+03	-2.7868E+03	-2.3539E+03	-2.1875E+03	-2.1233E+03	-3.2678E+03	-3.0607E+03	-2.2242E+03	-2.9708E+03
Best		-3.0047E+03	-3.2028E+03	-3.7732E+03	-2.1068E+03	-2.9924E+03	-2.8849E+03	-2.5174E+03	-2.3933E+03	-3.3592E+03	-3.2744E+03	-4.1376E+03	-4.1376E+03
STD		4.5283E+02	3.6950E+02	6.6467E+02	1.3411E+02	1.6730E+02	3.9855E+02	2.8876E+02	4.2458E+02	4.1710E+02	3.9286E+02	1.0526E+02	1.2856E+03
p-value		3.9442E-01	7.7926E-01	9.0927E-01	1.7657E-01	7.8607E-01	3.9466E-01	2.7939E-01	2.4719E-01	6.7574E-01	8.9796E-01	2.9101E-01	NaN
h	0	0	0	0	0	0	0	0	0	0	0	NaN	
Rank	8	6	4	12	5	7	10	11	1	2	9	3	
F9	Worst	0.0000E+00	4.2783E+01	0.0000E+00	4.2364E-02	3.8089E+01	4.4666E+00	1.2955E+01	5.2733E+01	1.9899E+00	1.8904E+01	0.0000E+00	0.0000E+00
Average		0.0000E+00	2.5869E+01	0.0000E+00	1.3584E-02	2.9275E+01	1.1166E+00	8.4629E+00	3.3331E+01	4.9748E+01	1.2437E+01	0.0000E+00	0.0000E+00
Best		0.0000E+00	1.1940E+01	0.0000E+00	1.7081E-11	1.9416E+01	0.0000E+00	5.9724E+00	2.3879E+01	0.0000E+00	6.9647E+00	0.0000E+00	0.0000E+00
STD		0.0000E+00	1.4869E+01	0.0000E+00	1.9999E-02	7.6693E+00	2.2333E+00	3.1025E+00	1.3112E+01	9.9496E+01	5.8862E+00	0.0000E+00	0.0000E+00
p-value		NaN	NaN	NaN	0	NaN	0	1	1	0	1	NaN	
h	NaN	1	NaN	0	1	0	1	1	0	1	0	NaN	
Rank	1	10	1	5	11	7	12	6	1	9	1	1	
F10	Worst	8.8818E-16	2.3168E+00	4.4409E-15	8.4300E-01	9.6530E+00	1.5099E-14	7.5055E-10	1.1551E+00	7.9936E-15	1.6462E+00	8.8818E-16	8.8818E-16
Average		8.8818E-16	1.1568E+00	4.4409E-15	2.1075E-01	4.7552E+00	9.7700E-15	4.3526E-10	2.8891E-01	5.3291E-15	7.0034E-01	8.8818E-	

Table 5. The results of the comparative methods and the proposed HRSA methods for F1–F13, Dim = 100.

Fun	Measure	Comparative Algorithms											
		AO	DMOA	WOA	SCA	DA	GWO	PSO	ALO	RSA	ROA	AOA	HRSA
F1	Worst	4.0782E-98	3.1718E+01	5.8226E-61	1.3866E+03	3.9007E+03	8.5638E-17	7.7635E-01	7.3039E+02	1.1830E-28	4.9992E+01	1.2631E-02	2.8449E-106
Average	1.0196E-98	2.2918E+01	1.4557E-61	8.8436E+02	1.6626E+03	4.9381E-17	4.7025E-01	2.2654E+02	3.2097E-29	3.5959E+01	5.8299E-03	7.1122E-107	
Best	3.0185E-123	9.3735E+00	1.0884E-73	1.3827E+02	3.6367E+02	5.5387E-18	1.4462E-01	5.4808E+01	2.3156E-30	2.5600E+01	1.8742E-10	7.5850E-149	
STD	2.0391E-98	1.0166E+01	2.9112E-61	5.6464E+02	1.6147E+03	3.3504E-17	2.6084E-01	3.3591E+02	5.7483E-29	1.0735E+01	6.1114E-03	1.4224E-106	
p-value	3.5590E-01	4.0661E-03	3.5589E-01	2.0260E-02	8.5130E-02	2.5688E-02	1.1290E-02	2.2608E-01	3.0682E-01	5.3690E-04	1.0502E-01	NaN	
h	0	1	0	1	0	1	1	0	0	1	0	NaN	
Rank	2	8	3	11	12	5	7	10	4	9	6	1	
F2	Worst	3.7433E-50	1.0812E+02	5.4876E-47	3.8977E-01	1.8393E+01	2.7882E-10	5.2010E+00	2.0965E+02	5.2108E-18	2.2379E+00	2.6869E-32	1.5364E-60
Average	9.3863E-51	4.1687E+01	1.4222E-47	2.0333E-01	1.2972E+01	1.8202E-10	3.3373E+00	1.0497E+02	3.5046E-18	1.5655E+00	6.7172E-33	3.8427E-61	
Best	1.2613E-57	1.1616E+01	1.6943E-51	4.6794E-02	8.4600E+00	6.8138E-11	1.2963E+00	2.3419E+01	1.5445E-18	1.0772E-60	6.7796E-60	1.3859E-75	
STD	1.8698E-50	4.4775E+01	2.7117E-47	1.5993E-01	4.1353E+00	9.3938E-11	1.6588E+00	9.4076E+01	1.5059E+08	5.0391E-01	1.3434E-32	7.6807E-61	
p-value	3.5413E-01	1.1190E-01	3.3491E-01	4.3918E-02	7.6223E-04	8.2145E-03	6.9291E-03	6.7126E-02	3.4872E-03	8.0222E-04	3.5592E-01	NaN	
h	0	0	0	1	1	1	1	0	1	1	0	NaN	
Rank	2	11	3	7	10	6	9	12	5	8	4	1	
F3	Worst	2.36803E-93	16951.21302	420591.1857	59991.69346	110891.9087	3.600143592	3296.378763	60132.02142	0.031346296	9792.661794	0.314587915	2.4468E-111
Average	5.92006E-94	14078.94053	259808.0922	53400.71341	78018.00548	1.18948309	2064.355479	37864.25151	0.00819798	7542.713664	0.160888346	6.1182E-112	
Best	3.6969E-112	9664.548042	164862.827	41214.97962	49573.50799	0.130123735	1085.580554	18446.14994	1.16188E-05	5634.887889	0.082118372	8.7014E-149	
STD	1.18401E-93	3249.194475	111450.2066	8824.145524	25173.64	1.627422681	914.7006548	17791.76731	0.015444974	2050.038886	0.108252379	1.2233E-111	
p-value	0.000130212	0.0003458995	1.93238E-05	0.000812637	0.194109297	0.004044426	0.005342487	0.329279344	0.000322532	0.024877921	NaN		
h	0	1	1	1	1	0	1	1	0	1	1	NaN	
Rank	2	8	11	10	11	5	6	9	3	7	4	1	
F4	Worst	6.8610E-60	3.1812E+01	9.0999E+01	7.4864E+01	4.3456E+01	5.5896E-03	6.5534E+00	3.6673E+01	8.9361E-06	4.8080E+01	8.2925E-02	1.7208E-54
Average	2.5424E-60	2.5308E+01	7.0297E+01	7.1699E+01	3.1073E+01	2.8448E-03	4.4164E+00	3.2017E+01	3.8129E-06	3.7531E+01	5.8325E-02	4.3063E-55	
Best	1.0303E-72	1.8835E+01	1.6764E+01	6.5484E+01	1.1452E+01	7.6601E-04	2.8777E+00	2.8534E+01	4.1312E-06	3.3232E+01	4.3176E-02	5.7683E-63	
STD	3.2744E-60	5.5607E+00	3.5899E+01	4.2435E+00	5.15349E+01	2.0889E-03	1.6731E+00	3.7908E+00	3.4653E-06	7.0710E+00	1.7231E-02	8.6015E-55	
p-value	3.5534E-01	9.8768E-05	7.8348E-03	4.4709E-08	6.7336E-03	3.4467E-02	1.8666E-03	2.7512E-06	7.0044E-02	4.1159E-05	5.0739E-04	NaN	
h	0	1	1	1	1	0	1	1	0	1	1	NaN	
Rank	2	8	12	10	11	5	6	9	3	7	4	1	
F5	Worst	1.6654E-02	4.2719E+04	4.8760E+01	2.2741E+07	4.0217E+05	4.8697E+01	1.1191E+03	3.6622E+04	4.6787E+01	2.1306E+03	4.8918E+01	4.5067E+01
Average	6.2950E-03	1.4282E+04	4.8363E+01	1.4207E+07	1.4645E+05	4.8074E+01	6.9694E+02	2.6873E+04	4.6359E+01	1.1988E+03	4.8877E+01	1.9759E+01	
Best	1.8375E-04	9.0197E+02	4.8283E+01	1.2060E+06	4.5968E+04	4.7080E+01	4.5280E+00	1.7436E+04	4.5956E+01	6.4804E+02	4.8824E+01	3.4406E-01	
STD	7.4526E-03	1.9583E+04	2.1777E-01	9.1797E+06	2.1082E+05	7.5301E-01	2.9226E+02	8.5652E+03	3.8440E-01	6.4881E+02	4.0631E-02	1.8644E+01	
p-value	7.8400E-02	1.9548E-01	2.1475E-02	2.1243E-02	1.3727E-01	2.2949E-02	3.5979E-03	7.6452E-04	2.9069E-02	1.0925E-02	2.0489E-02	NaN	
h	0	0	1	1	0	1	1	1	1	1	1	NaN	
Rank	1	7	11	12	8	4	6	9	3	10	5	2	
F6	Worst	7.4588E-03	4.0782E+01	2.8398E+00	4.1424E+03	2.1428E+03	4.6882E+00	1.5872E+00	3.2536E+02	5.2899E-01	4.0303E+01	8.8853E+00	5.0952E-02
Average	2.0732E-03	2.4752E+01	2.5240E+00	1.7732E+03	1.5093E+03	3.5812E+00	7.6906E-01	2.1992E+02	4.3057E-01	2.8780E+01	8.5206E+00	2.1823E-02	
Best	8.0662E-05	1.5665E+01	2.1190E+00	3.7947E+02	6.3968E+02	2.6152E+00	1.8089E-01	1.0497E+02	2.6481E-01	1.2823E+01	8.1968E+00	8.4905E-04	
STD	3.5956E-03	1.1063E+01	3.2233E-01	1.7914E+03	6.3437E+02	9.3909E-01	6.0022E+00	9.0506E+01	1.2080E+01	1.2291E+01	3.2283E-01	2.4930E-02	
p-value	1.6789E-01	4.2335E-03	4.5982E-06	9.5071E-02	3.1321E-03	2.6630E-04	4.7309E-02	2.8246E-03	5.6882E-04	3.3981E-03	3.2074E-09	NaN	
h	0	1	1	0	1	1	1	1	1	1	1	NaN	
Rank	1	9	5	12	11	4	7	10	3	8	6	2	
F7	Worst	8.2152E-04	7.8054E-01	6.5132E-03	2.8601E+00	3.8227E-01	8.1132E-03	1.2213E+01	1.9458E+00	3.5460E-03	1.3288E+00	4.7987E-03	1.1384E-04
Average	5.5749E-04	6.9759E-01	3.3080E-03	2.6546E+00	2.5167E-01	4.0133E-03	7.1154E-00	1.3356E+00	2.4778E-03	9.4944E-01	1.8265E-03	7.3664E-05	
Best	2.2004E-04	5.5688E-01	1.2007E-03	2.4594E+00	1.3103E-01	1.6145E-03	1.7505E+00	7.0578E-01	1.4960E-03	6.5989E-01	4.8190E-04	2.4998E-05	
STD	2.9741E-04	9.7832E-02	2.5103E-03	1.7191E-01	1.1463E-01	2.9558E-03	4.4619E+00	5.0639E-01	8.4777E-04	3.2501E-01	1.9976E-03	4.0096E-05	
p-value	2.5556E-01	7.5583E-06	3.9132E-01	7.6873E-08	4.7787E-03	2.6615E-01	1.8871E-02	1.8873E-03	5.7030E-01	1.1918E-03	1.2985E-01	NaN	
h	0	1	0	1	1	0	1	1	0	1	0	NaN	
Rank	2	8	5	12	11	6	4	10	3	9	3	1	
F8	Worst	-5.4374E-03	-1.1297E+04	-3.1741E+03	-4.5531E+03	-6.7763E+03	-6.5491E+03	-4.5026E+03	-9.0295E+03	-1.3621E+04	-1.0546E+04	-4.6339E+03	-1.3186E+04
Average	-5.8281E-03	-1.2227E+04	-6.8610E+03	-4.9929E+03	-7.9585E+03	-8.9655E+03	-5.4962E+03	-9.0590E+03	-1.4362E+04	-1.2700E+04	-5.6555E+03	-1.7042E+04	
Best	-6.0837E-03	-1.3238E+04	-1.0029E+04	-5.2303E+03	-9.1043E+03	-1.0478E+04	-6.0557E+03	-9.1475E+03	-1.5873E+04	-1.4023E+04	-6.5546E+03	-2.0457E+04	
STD	2.9576E-02	9.1368E+02	3.1291E+03	3.1477E+02	9.5197E+02	1.7889E+02	7.2763E+02	5.9039E+01	1.0288E+03	1.5246E+03	7.9497E+02	3.7487E+03	
p-value	5.3540E-01	1.6567E-02	5.8778E-03	2.7973E-01	5.2715E-01	2.8720E-01	4.2813E-01	2.0974E-01	3.8734E-01	1.5321E-02	4.8344E-01	NaN	
h	0	1	1	0	0	0	0	0	1	1	0	NaN	
Rank	9	4	8	12	7	6	11	5	2	3	10	1	
F9	Worst	0.0000E+00	1.7042E+02	0.0000E+00	2.4160E+02	3.2101E+02	1.0984E+01	2.4610E+02	1.6151E+02	0.0000E+00	1.6151E+02	0.0000E+00	0.0000E+00
Average	0.0000E+00	1.3938E+02	0.0000E+00	1.5839E+02	2.4885E+02	4.4390E+02	1.8299E+02	1.4673E+02	0.0000E+00	1.2378E+02	0.0000E+00	0.0000E+00	
Best	0.0000E+00	1.6126E+02	0.0000E+00	7.4822E+01	1.2753E+02	1.3074E-12	1.4064E+02	1.2878E+02	0.0000E+00	9.6050E+01	0.0000E+00	0.0000E+00	
STD	0.0000E+00	2.2879E+01	0.0000E+00	6.9738E+01	8.5115E+01	5.4063E+00	4.6890E+01	1.4200E+01	0.0000E+00	2.7639E+01	0.0000E+00	0.0000E+00	
p-value	NaN	1.8591E-05	NaN	1.0130E-03	5.6744E-03	2.3326E-04	8.3516E-04	NaN	1.0816E-04	NaN	NaN	NaN	
h	NaN	1	NaN	1	1	0	1	1	NaN	1	NaN	NaN	
Rank	1	8	1	10	12	6	7	11	9	1	7	1	
F10	Worst	8.8818E-16	7.2668E+00	7.9936E-15	2.0531E+01	8.9559E+00	4.3052E-09	2.2928E+00	1.5690E+01	3.2863E-14	1.8274E+01	8.9706E-14	8.8818E-16
Average	8.8818E-16	6.3388E+00	2.6645E-15	2.0518E+01	7.2218E+00	1.7663E-09	1.9853E+00	1.3964E+01	2.9310E-14	1.3315E+01	3.0198E-14	8.8818E-16	
Best	8.8818E-16	5.0825E+00											

Table 6 shows the results of the comparative methods and the proposed HRSA methods for fixed dimensional functions (F14–F23). As shown in the table, the suggested technique outperformed the existing comparison methods in these circumstances. This result revealed the enhanced approach's ability to overcome the critical weaknesses of the old method. According to the Wilcoxon signed-rank test, the proposed HRSA overwhelmed almost all the comparative methods in all the given cases. For example, the proposed HRSA significantly overwhelmed DMOA, SCA, DA, ALO, RSA, and ROA in F14. Additionally, it gave a significant improvement compared to AO, DMOA, and SCA in F23. According to the Friedman ranking test, the proposed HRSA ranked as the first-best method, followed by the RSA method as the second-best method, the ROA as the third-best method, the GWO as the third-best method, the ALO as the fifth-best method, the DMOA as the sixth-best method, the GWO as the seventh-best method, the AO as the eighth-best method, the DA as the ninth-best method, the WOA as the tenth-best method, the SCA as the eleventh-best method, and finally, the AOA as the twentieth-best method. We came to the conclusion that the proposed HRSA can find better solutions than previous methods when the dimensions are fixed.

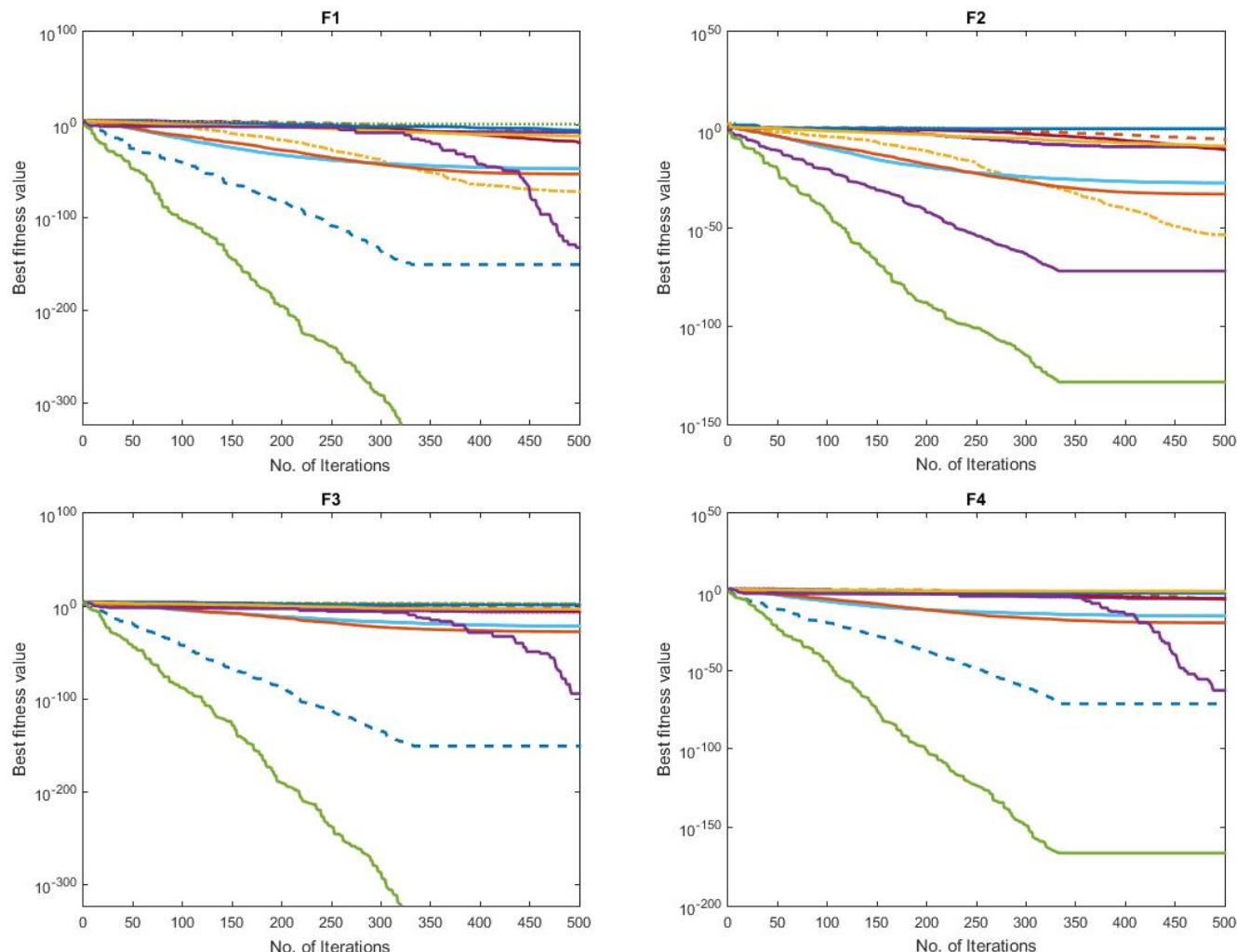
Table 6. The results of the comparative methods and the proposed HRSA methods for fixed dimensional functions (F14–F23).

Fun	Measure	Comparative Algorithms											
		AO	DMOA	WOA	SCA	DA	GWO	PSO	ALO	RSA	ROA	AOA	HRSA
F14	Worst	1.2671E+01	9.9800E-01	1.0763E+01	2.9821E+00	1.9920E+00	1.2671E+01	8.8408E+00	6.9033E+00	9.9800E-01	1.2671E+01	1.2671E+01	9.9800E-01
	Average	4.9092E+00	9.9800E-01	4.6720E+00	1.5058E+00	1.2465E+00	4.6607E+00	6.3958E+00	3.4654E+00	9.9800E-01	1.0248E+01	1.0010E+01	9.9800E-01
	Best	1.9920E+00	9.9800E-01	9.9800E-01	9.9846E-01	9.9800E-01	9.9800E-01	9.9800E-01	9.9800E-01	9.9800E-01	2.9821E+00	2.9821E+00	9.9800E-01
	STD	5.1952E+00	1.2820E-16	4.6790E+00	9.8443E-01	4.9701E-01	5.4010E+00	3.7126E+00	2.6033E+00	2.2204E-16	4.8442E+00	4.7070E+00	1.2820E-16
	p-value	1.8345E-01	8.7691E-03	1.4881E-01	1.2260E-02	1.0122E-02	1.7440E-01	2.5362E-01	4.8667E-02	8.7691E-03	8.7691E-03	9.4612E-01	NaN
	h	0	1	0	1	1	0	0	1	1	1	0	NaN
F15	Rank	9	3	8	5	4	7	10	6	2	12	11	1
F16	Worst	1.6162E-03	2.0364E-02	8.5450E-04	1.5480E-03	2.0363E-02	2.0363E-02	1.0751E-03	1.5316E-03	1.6133E-03	1.2232E-03	4.8245E-02	5.4450E-04
	Average	1.1160E-03	5.6279E-03	5.3753E-04	1.1086E-03	6.6307E-03	5.4880E-03	9.4545E-04	1.1987E-03	7.1362E-04	1.0923E-03	2.4085E-02	4.1911E-04
	Best	6.9512E-04	5.3740E-04	3.1440E-04	5.5412E-04	1.6554E-04	3.2074E-04	7.2990E-04	7.6864E-04	3.2389E-04	6.9969E-04	4.8440E-04	3.2524E-04
	STD	3.8035E-04	9.8253E-03	2.5721E-04	4.9369E-04	9.1594E-03	9.9180E-03	1.4908E-04	3.1653E-04	6.0669E-04	2.6174E-04	2.1027E-02	9.1516E-05
	p-value	1.1888E-02	3.9415E-01	4.5309E-02	9.8187E-01	2.7425E-01	4.1222E-01	4.3345E-01	7.4953E-01	3.0403E-01	9.2164E-01	7.1625E-02	NaN
	h	1	0	1	0	0	0	0	0	0	0	0	NaN
F17	Rank	7	10	2	6	11	9	4	8	3	12	1	1
F18	Worst	-1.0306E+00	-1.0316E+00										
	Average	-1.0313E+00	-1.0316E+00	-1.0315E+00	-1.0316E+00								
	Best	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00
	STD	4.4362E-04	7.4564E-15	9.7513E-09	9.0048E-05	1.5128E-05	4.4567E-08	6.6134E-03	1.8769E-13	1.2820E-16	0.0000E+00	1.5246E-07	1.2820E-16
	p-value	4.8620E-02	4.2244E-02	4.4387E-02	4.2563E-02	4.2225E-02	4.2224E-02	4.2224E-02	4.2224E-02	4.2224E-02	4.2224E-02	4.2230E-02	NaN
	h	1	1	1	1	1	1	1	1	1	1	1	NaN
F19	Rank	11	4	6	10	9	7	12	5	3	1	8	1
F20	Worst	3.0717E+00	3.0000E+00	3.0022E+00	3.0003E+00	3.0000E+00	8.4000E+01	3.0000E+00	3.0000E+00	3.0009E+01	3.0000E+00	9.3087E+01	3.0000E+00
	Average	3.0363E+00	3.0000E+00	3.0007E+00	3.0001E+00	3.0000E+00	2.3250E+01	3.0000E+00	3.0000E+00	1.0151E+01	3.0000E+00	4.8067E+01	3.0000E+00
	Best	3.0066E+00	3.0000E+00	3.0001E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.1441E+00	3.0000E+00	3.0000E+00	3.0000E+00
	STD	3.3175E-02	3.5171E-13	9.9674E-04	1.3095E-04	1.5141E-10	4.0500E-01	2.8320E-15	5.9994E-13	1.3294E+01	5.1279E-16	5.1876E+01	1.7578E-15
	p-value	3.2559E-01	3.2233E-01	3.2338E-01	3.2334E-01	5.6139E-01	3.2333E-01	3.2333E-01	3.2333E-01	3.2333E-01	3.2333E-01	2.0653E-01	NaN
	h	0	0	0	0	0	0	0	0	0	0	0	NaN
F21	Rank	10	5	9	11	7	8	1	6	1	4	12	1
F22	Worst	-3.8552E+00	-3.8562E+00	-3.7519E+00	-3.8549E+00	-3.8562E+00	-3.8556E+00	-3.8562E+00	-3.8562E+00	-3.8562E+00	-3.8562E+00	-3.85415E+00	-3.85628E+00
	Average	-3.8557E+00	-3.8562E+00	-3.85251E+00	-3.85520E+00	-3.8562E+00	-3.85589E+00	-3.8562E+00	-3.8562E+00	-3.8562E+00	-3.8562E+00	-3.85450E+00	-3.85628E+00
	Best	-3.85627E+00	-3.85628E+00	-3.85555E+00	-3.85547E+00	-3.85628E+00	-3.85627E+00	-3.85628E+00	-3.85628E+00	-3.85628E+00	-3.85628E+00	-3.85523E+00	-3.85628E+00
	STD	4.3471E-03	1.2362E-09	4.9029E-02	2.8013E-03	7.9323E-05	3.6209E-03	0.0000E+00	5.3811E-13	3.6260E-16	1.8733E-01	4.9600E-03	0.0000E+00
	p-value	3.0902E-02	2.8796E-02	5.4899E-02	3.3516E-02	2.8824E-02	3.0430E-02	2.8796E-02	2.8796E-02	2.8796E-02	2.8796E-02	3.7043E-02	NaN
	h	1	1	0	1	1	1	1	1	1	1	1	NaN
F23	Rank	9	4	8	7	6	11	5	10	1	12	3	3

Table 6. Cont.

Fun	Measure	Comparative Algorithms											
		AO	DMOA	WOA	SCA	DA	GWO	PSO	ALO	RSA	ROA	AOA	HRSA
F23	Worst	-1.0512E+01	-4.8806E+00	-1.8585E+00	-9.4646E-01	-2.4217E+00	-2.8711E+00	-2.8066E+00	-2.4217E+00	-3.85354E+00	-1.2013E+00	-1.0536E+01	
	Average	-1.0523E+01	-6.9426E+00	-4.9871E+00	-3.2865E+00	-4.7987E+00	-8.5059E+00	-8.6201E+00	-8.6040E+00	-8.5077E+00	-7.1859E+00	-4.2224E+00	-1.0536E+01
	Best	-1.0535E+01	-9.3471E+00	-1.0536E+01	-4.6174E+00	-1.0528E+01	-1.0535E+01	-1.0536E+01	-1.0536E+01	-1.0536E+01	-1.0536E+01	-7.3080E+00	-1.0536E+01
	STD	1.0420E-02	2.2857E+00	3.9644E+00	1.6939E+00	3.8764E+00	4.0562E+00	3.8326E+00	3.8649E+00	4.0573E+00	3.8688E+00	2.5257E+00	4.0454E-11
	p-value	2.0261E-02	1.9951E-02	4.2554E-01	4.2320E-02	3.7748E-01	5.2688E-01	4.8060E-01	4.8724E-01	5.2649E-01	9.1729E-01	1.6135E-01	NaN
	h	1	1	0	1	0	0	0	0	0	0	0	NaN
	Rank	2	8	9	12	10	6	3	4	5	7	11	1
Mean ranking		7.2000E+00	6.0000E+00	8.1000E+00	9.6000E+00	7.3000E+00	6.7000E+00	5.4000E+00	5.7000E+00	3.7000E+00	5.4000E+00	1.0800E+01	1.4000E+00
Final ranking		8	6	10	11	9	7	3	5	2	3	12	1

The convergence behaviors of the comparative optimization methods employing the evaluated twenty-three benchmark functions are shown in Figure 3. The optimization history and trajectory during the evolution processes are depicted in these diagrams. In comparison to the previous comparative approaches, it is evident that the proposed HRSA obtains optimum solutions in all of the studied cases and converges to the optimal solution smoothly. Furthermore, the observed convergence tendency is appealing and reinforces the original method's primary flaws. As a result, the primary issues with the original approaches are addressed, as seen in the figures. Figure 4 shows the distribution values of the given position during the optimization process. It is clear that the proposed method generates higher distribution values during the improvement (optimization) process. This evidences that modifying the original method achieved new results that satisfied the proposed idea.

**Figure 3.** Cont.

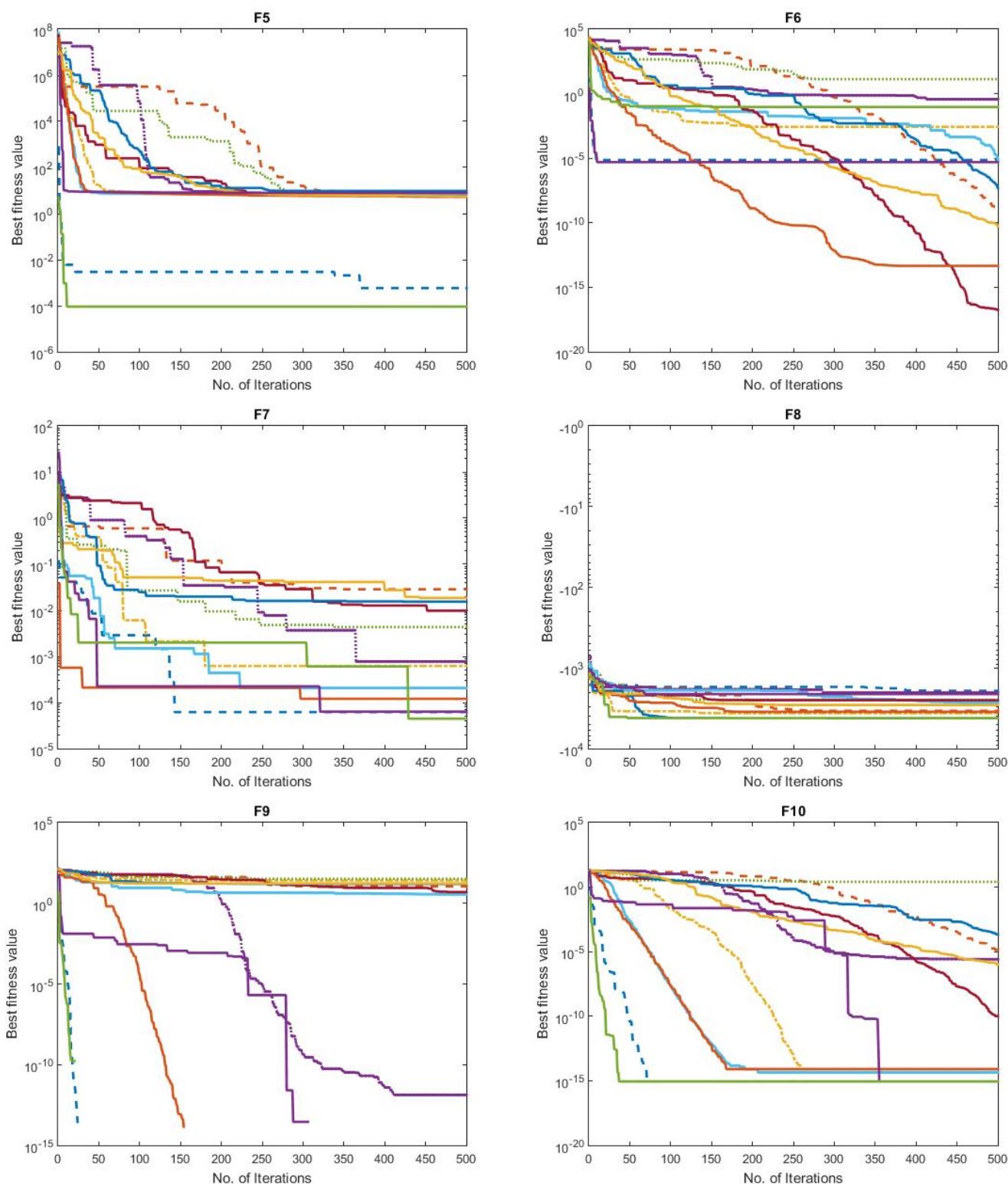


Figure 3. Cont.

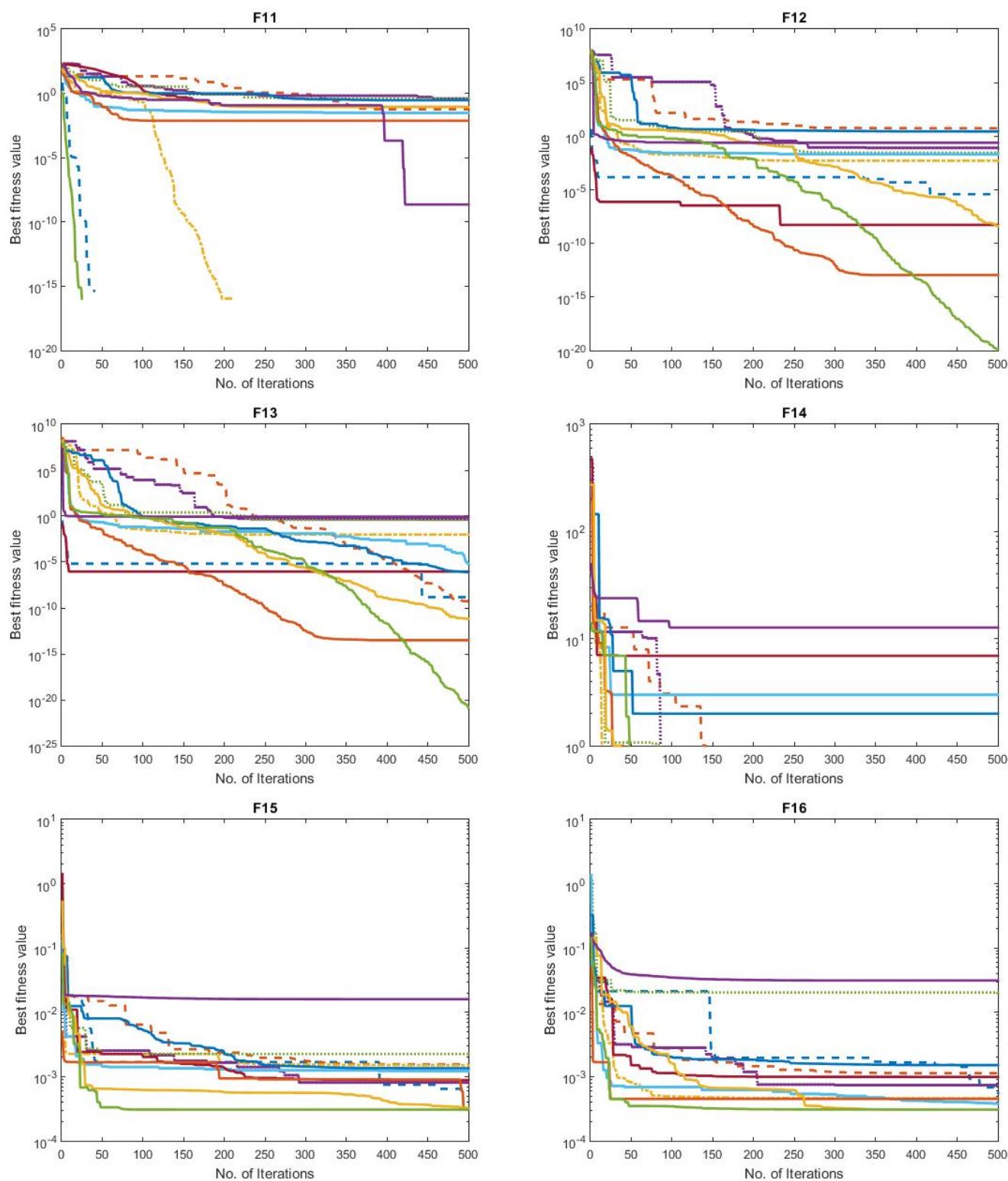


Figure 3. Cont.

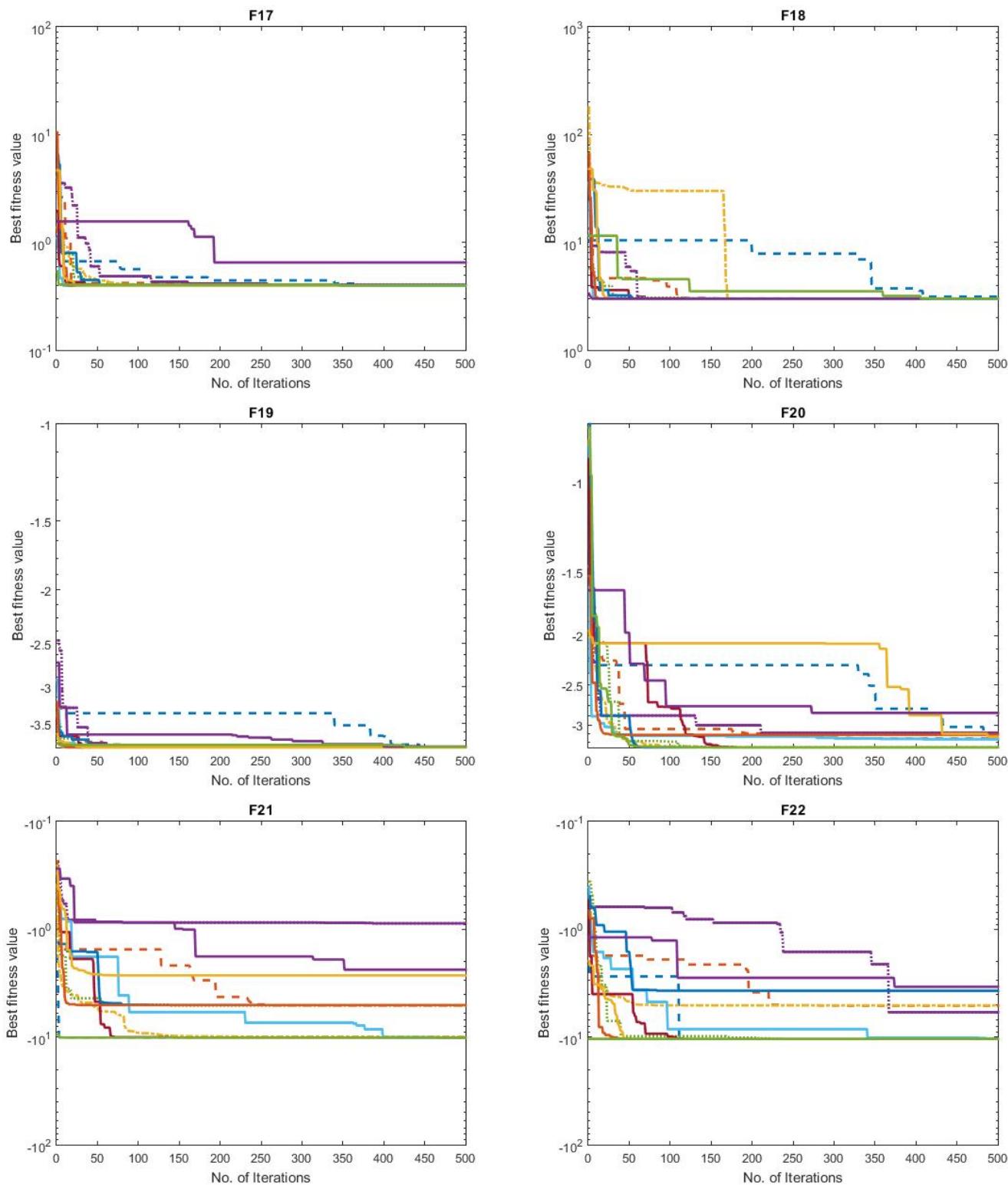


Figure 3. Cont.

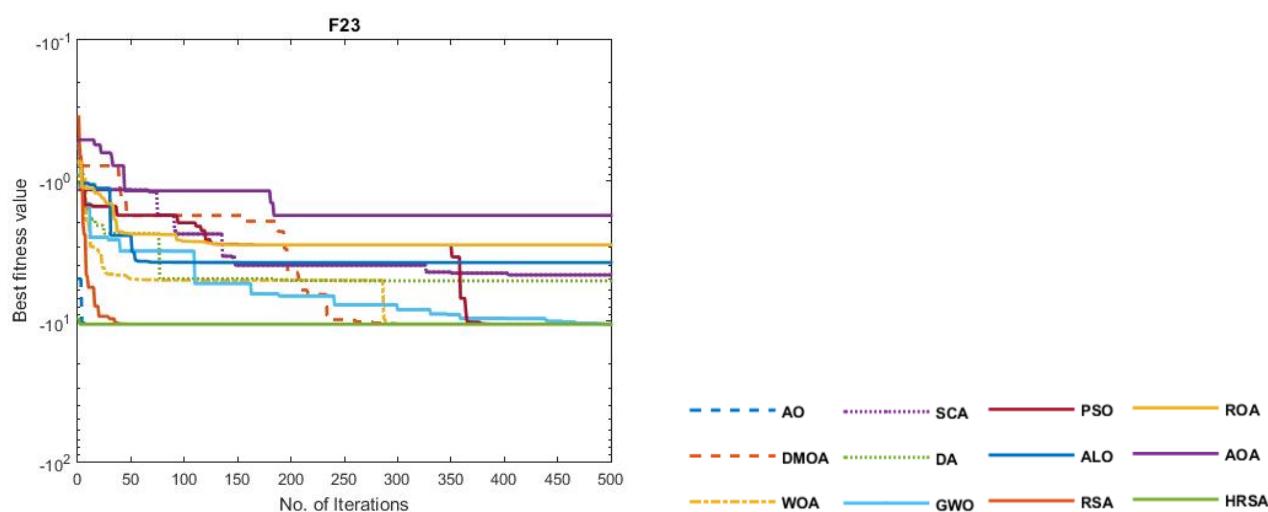


Figure 3. Convergence curves of HRSA and other methods for F1–F23.

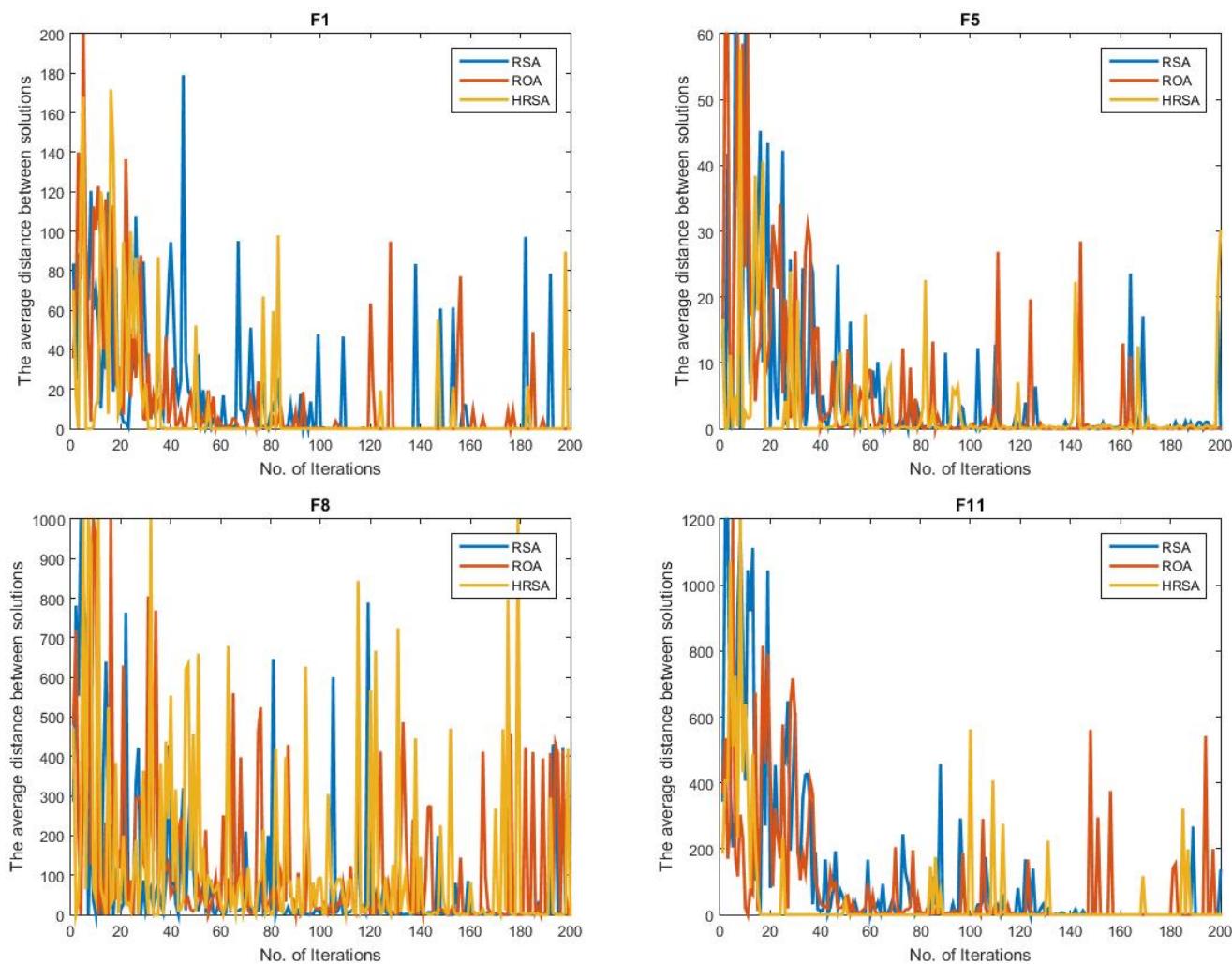


Figure 4. The distribution values of the given position during the optimization process.

3.3. Data Clustering: Experiments

In this section, the proposed method is tested further for solving the data clustering problems.

Eight problems are used with different characterizations shown in Table 7: the number of features, objects, and clusters. These datasets were taken from the UCI repository. They are always used in the domain of machine learning and clustering algorithms. The results of the proposed method are compared with other methods such as Aquila Optimizer (AO) [32], Particle Swarm Optimizer (PSO) [33], Grey Wolf Optimizer (GWO) [34], African Vultures Optimization Algorithm (AVOA) [35], Whale Optimization Algorithm (WOA) [36], Reptile Search Algorithm (RSA) [27], Remora Optimization Algorithm (ROA) [28], Arithmetic Optimization Algorithm (AOA) [37], and the proposed HRSA method. All the tested methods are tuned according to the original paper and its parameters. Each method was executed 30 times using 50 solutions and 1000 iterations.

Table 7. UCI benchmark datasets.

Dataset	Features No.	Instances No.	Classes No.
Cancer	9	683	2
CMC	10	1473	3
Glass	9	214	7
Iris	4	150	3
Seeds	7	210	3
Heart	13	270	2
Vowels	6	871	3
Water	13	178	3

Table 8 shows the obtained results of the proposed HRSA method and other comparative clustering methods using eight datasets. This table presents and measures the results of nine comparative algorithms in terms of the worst, best, average fitness function values. Additionally, the standard deviation values are presented to show the variate of the presented results. The Wilcoxon signed-rank test was also used to show the degree of improvements achieved by the proposed method compared to other comparative methods. Then, the Friedman ranking test was used to find the rank of the tested methods for all the used datasets.

In Table 8, the proposed HRSA achieved better results in almost all the tested problems compared to the other comparative methods. The proposed method proved its performance in solving the given problems. It is the best method overall, compared to the other methods. The proposed method reaches optimal solutions in almost all the tested datasets, and its solutions are better than the previous solutions. According to the Wilcoxon signed-rank test, the proposed HRSA overwhelmed almost all the comparative methods. For example, the proposed HRSA significantly overwhelmed the cancer dataset's ROA, GWO, PSO, and AO. Additionally, it achieved significant improvement compared to ROA, WOA, GWO, PSO, and AO in the class dataset. According to the Friedman ranking test, the proposed HRSA is ranked the first-best method, followed by the PSO method as the second best method, the GWO as the third-best method, the AO as the fourth-best method, the ROA as the fifth-best method, the AOA and RSA as the sixth-best methods, the WOA as the eighth best method, and finally, the AVOA as the ninth-best method. The final results are explained in Figure 5.

Figure 6 shows the convergence behavior of the comparative optimization algorithms using the eight tested data clustering problems. These figures are used to show the optimization history and the trajectory during the evolution processes. It is clear that the proposed HRSA reaches the optimal solutions in all the tested problems and converges to the optimal solution smoothly compared to the other comparative methods. Moreover, the convergence behavior observed is attractive and strengthens the main weaknesses of the original method. Thus, the main problems of the original methods are solved as given in the given figures.

Table 8. The comparative results of HRSA and other clustering methods using eight datasets.

Dataset	Metric	Comparative Algorithms								
		AO	PSO	GWO	AVOA	WOA	RSA	ROA	AOA	HRSA
Cancer	Worst	2.9098E+03	1.6727E+03	2.9680E+03	3.4670E+03	3.5238E+03	3.5050E+03	3.5050E+03	3.5050E+03	7.1703E+02
	Average	2.7388E+03	1.2148E+03	2.6393E+03	3.3097E+03	3.4419E+03	3.2415E+03	3.2415E+03	3.2415E+03	4.9848E+02
	Best	2.6050E+03	6.7650E+02	2.3894E+03	3.0439E+03	3.3555E+03	2.7880E+03	2.7880E+03	2.7880E+03	2.5294E+02
	STD	1.2285E+02	4.1186E+02	2.7025E+02	2.1505E+02	7.5799E+01	3.0883E+02	3.0883E+02	3.0883E+02	1.9627E+02
	p-value	9.6175E-03	2.1793E-05	1.1166E-02	6.9577E-01	1.9649E-01	1.0000E+00	1.6244E-07	1.0000E+00	NaN
	h	1	1	1	0	0	0	1	0	NaN
	Rank	4	2	3	8	9	5	5	5	1
	Worst	3.3274E+02	1.0104E+02	3.1034E+02	3.3504E+02	3.3443E+02	3.3470E+02	3.3470E+02	3.3470E+02	8.1049E+01
	Average	3.2908E+02	8.4904E+01	3.0460E+02	3.3428E+02	3.3407E+02	3.3332E+02	3.3332E+02	3.3332E+02	6.7316E+01
	Best	3.2649E+02	6.0653E+01	3.0189E+02	3.3292E+02	3.3340E+02	3.3199E+02	3.3199E+02	3.3199E+02	5.0947E+01
CMC	STD	2.3268E+00	1.4951E+01	3.2968E+00	8.1148E-01	4.6832E-01	1.1409E+00	1.1409E+00	1.1409E+00	1.3054E+01
	p-value	6.3870E-03	3.0919E-10	7.7964E-08	1.6589E-01	2.1389E-01	1.0000E+00	6.1264E-11	1.0000E+00	NaN
	h	1	1	1	0	0	0	1	0	NaN
	Rank	4	2	3	9	8	5	5	5	1
	Worst	3.1433E+01	1.6129E+01	2.9980E+01	3.5149E+01	3.4978E+01	3.5062E+01	4.3268E+00	3.5062E+01	3.5062E+01
	Average	3.0876E+01	1.2067E+01	2.9199E+01	3.4583E+01	3.3710E+01	3.4938E+01	1.3341E+00	3.4938E+01	3.4938E+01
	Best	3.0338E+01	9.9866E+00	2.7810E+01	3.4088E+01	3.2553E+01	3.4822E+01	0.0000E+00	3.4822E+01	3.4822E+01
	STD	5.0092E-01	2.3606E+00	8.8269E-01	4.4374E-01	9.5919E-01	1.1264E-01	1.7931E+00	1.1264E-01	1.1264E-01
	p-value	1.0662E-07	2.1913E-08	5.2248E-07	1.2103E-01	2.1695E-02	1.0000E+00	1.1771E-10	1.0000E+00	1.0000E+00
	h	1	1	1	0	1	0	1	0	NaN
Glass	Rank	4	2	3	6	5	7	1	7	NaN
	Worst	2.1833E+01	6.5278E+00	2.1616E+01	2.4896E+01	2.4487E+01	2.4388E+01	2.4388E+01	2.4388E+01	5.5340E+00
	Average	2.0176E+01	5.5161E+00	1.6522E+01	2.4451E+01	2.3746E+01	2.3636E+01	2.3636E+01	2.3636E+01	2.8836E+00
	Best	1.8980E+01	4.1415E+00	1.3271E+01	2.4156E+01	2.3303E+01	2.3056E+01	2.3056E+01	2.3056E+01	1.5958E+00
	STD	1.1616E+00	9.9876E-01	3.0780E+00	2.9205E-01	5.3502E-01	6.2257E-01	6.2257E-01	6.2257E-01	1.7265E+00
	p-value	3.7448E-04	5.5402E-10	9.7090E-04	2.9211E-02	7.7219E-01	1.0000E+00	6.4135E-09	1.0000E+00	NaN
	h	1	1	1	1	0	0	1	0	NaN
	Rank	4	2	3	9	8	5	5	5	1
	Worst	4.0905E+01	1.9643E+01	3.9253E+01	5.0452E+01	5.0731E+01	5.0011E+01	5.0011E+01	5.0011E+01	1.0401E+01
	Average	3.9847E+01	1.3700E+01	3.7736E+01	4.9855E+01	4.9928E+01	4.8814E+01	4.8814E+01	4.8814E+01	8.5169E+00
Iris	Best	3.8646E+01	5.7750E+00	3.6119E+01	4.9054E+01	4.7994E+01	4.7821E+01	4.7821E+01	4.7821E+01	6.6314E+00
	STD	8.7573E-01	5.2011E+00	1.1579E+00	5.2647E-01	1.1100E+00	9.0900E-01	9.0900E-01	9.0900E-01	1.3364E+00
	p-value	2.4678E-07	4.1207E-07	1.5759E-07	5.7550E-02	1.2069E-01	1.0000E+00	1.1892E-11	1.0000E+00	NaN
	h	1	1	1	0	0	0	1	0	NaN
	Rank	4	2	3	8	9	5	5	5	1
	Worst	4.0905E+01	1.9643E+01	3.9253E+01	5.0452E+01	5.0731E+01	5.0011E+01	5.0011E+01	5.0011E+01	1.0401E+01
	Average	3.9847E+01	1.3700E+01	3.7736E+01	4.9855E+01	4.9928E+01	4.8814E+01	4.8814E+01	4.8814E+01	8.5169E+00
	Best	3.8646E+01	5.7750E+00	3.6119E+01	4.9054E+01	4.7994E+01	4.7821E+01	4.7821E+01	4.7821E+01	6.6314E+00
	STD	8.7573E-01	5.2011E+00	1.1579E+00	5.2647E-01	1.1100E+00	9.0900E-01	9.0900E-01	9.0900E-01	1.3364E+00
	p-value	2.4678E-07	4.1207E-07	1.5759E-07	5.7550E-02	1.2069E-01	1.0000E+00	1.1892E-11	1.0000E+00	NaN
Seeds	h	1	1	1	0	0	0	1	0	NaN
	Rank	4	2	3	6	5	7	7	7	1
	Worst	1.4705E+03	5.3201E+02	1.2471E+03	1.6629E+03	1.6207E+03	1.6982E+03	1.6982E+03	1.6982E+03	6.6389E+01
	Average	1.4064E+03	2.5262E+02	1.0091E+03	1.5678E+03	1.5659E+03	1.6325E+03	1.6325E+03	1.6325E+03	2.7628E+01
	Best	1.2522E+03	0.0000E+00	8.0036E+02	1.4391E+03	1.4899E+03	1.5087E+03	1.5087E+03	1.5087E+03	0.0000E+00
	STD	8.8824E+01	2.1268E+02	1.5920E+02	8.0923E+01	5.2544E+01	7.4898E+01	7.4898E+01	7.4898E+01	3.1474E+01
	p-value	2.4418E-03	7.8363E-07	4.6791E-05	2.2569E-01	1.4195E-01	1.0000E+00	7.6147E-11	1.0000E+00	NaN
	h	1	1	1	0	0	0	1	0	NaN
	Rank	4	2	3	6	5	7	7	7	1
	Worst	1.3952E+02	1.5133E+01	1.3594E+02	1.5347E+02	1.5231E+02	1.5336E+02	1.5336E+02	1.5336E+02	1.8987E+01
Vowels	Average	1.3832E+02	1.0040E+01	1.3417E+02	1.5304E+02	1.5175E+02	1.5287E+02	1.5287E+02	1.5287E+02	1.5031E+01
	Best	1.3666E+02	0.0000E+00	1.3111E+02	1.5275E+02	1.5121E+02	1.5235E+02	1.5235E+02	1.5235E+02	9.6841E+00
	STD	1.2751E+00	6.2258E+00	1.8819E+00	2.8123E-01	4.5754E-01	3.9920E-01	3.9920E-01	3.9920E-01	4.2281E+00
	p-value	8.6484E-09	2.3486E-11	2.1235E-08	4.4898E-01	3.4024E-03	1.0000E+00	1.4476E-12	1.0000E+00	NaN
	h	1	1	1	0	1	0	1	0	NaN
	Rank	4	1	3	9	5	6	6	6	2
	Worst	3.4751E+03	3.9940E+03	2.9327E+03	4.0307E+03	3.9606E+03	3.9940E+03	3.9940E+03	3.9940E+03	1.6141E+03
	Average	3.2725E+03	3.9170E+03	2.5840E+03	3.9325E+03	3.8541E+03	3.9170E+03	3.9170E+03	3.9170E+03	1.2385E+03
	Best	3.1515E+03	3.8401E+03	2.3275E+03	3.7908E+03	3.7249E+03	3.8401E+03	3.8401E+03	3.8401E+03	8.8523E+02
	STD	1.3674E+02	6.0335E+01	2.3210E+02	9.2329E+01	1.0214E+02	6.0335E+01	6.0335E+01	6.0335E+01	2.9581E+02
Wine	p-value	1.1140E-05	4.3418E-08	1.6402E-06	7.6057E-01	2.6979E-01	1.0000E+00	1.2556E-10	1.0000E+00	NaN
	h	1	1	1	0	0	0	1	0	NaN
	Rank	4	6	3	10	5	6	6	6	1
	Mean ranking	4.0000E+00	2.3750E+00	3.0000E+00	8.1250E+00	6.7500E+00	5.7500E+00	5.0000E+00	5.7500E+00	1.1429E+00
Final ranking		4	2	3	9	8	6	5	6	1

Figure 7 shows the clustering plot images using the eight tested datasets. This figure is used to display the clustered objects and the distribution of the objective over the specified area. The given results clearly show that the given objects are distributed in a similar area, and each area presents a similar topic. The clustering accuracy is recognized, which

proves the performance of the proposed HRSA method. The execution time ranking is given in Figure 8, which clearly shows that the proposed method has a good execution time compared to other comparative methods.

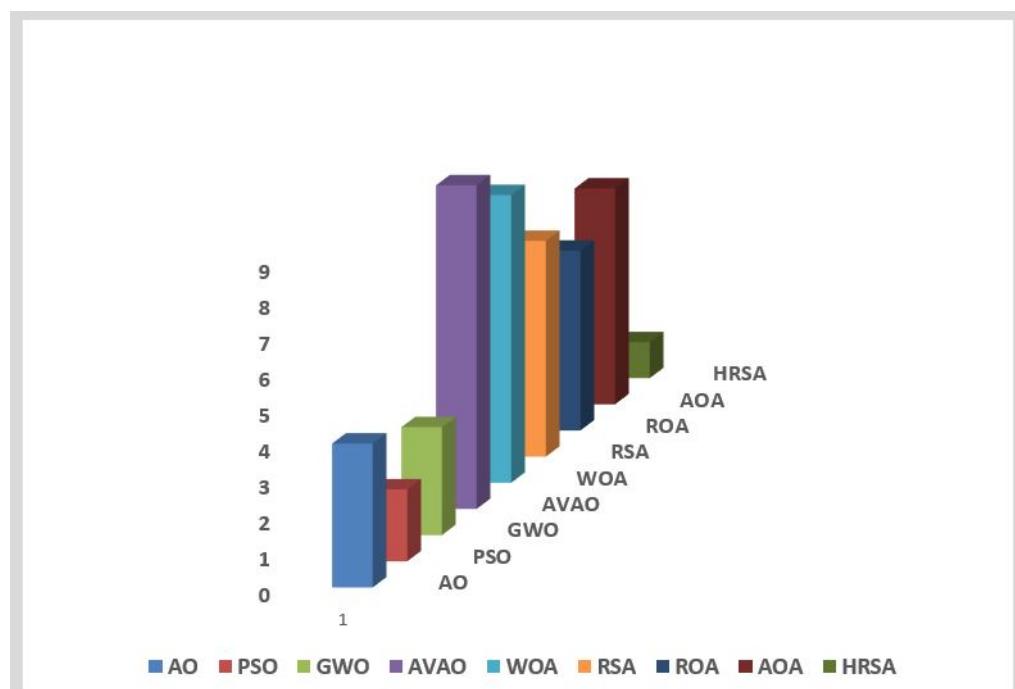


Figure 5. Execution time ranking for the benchmark functions.

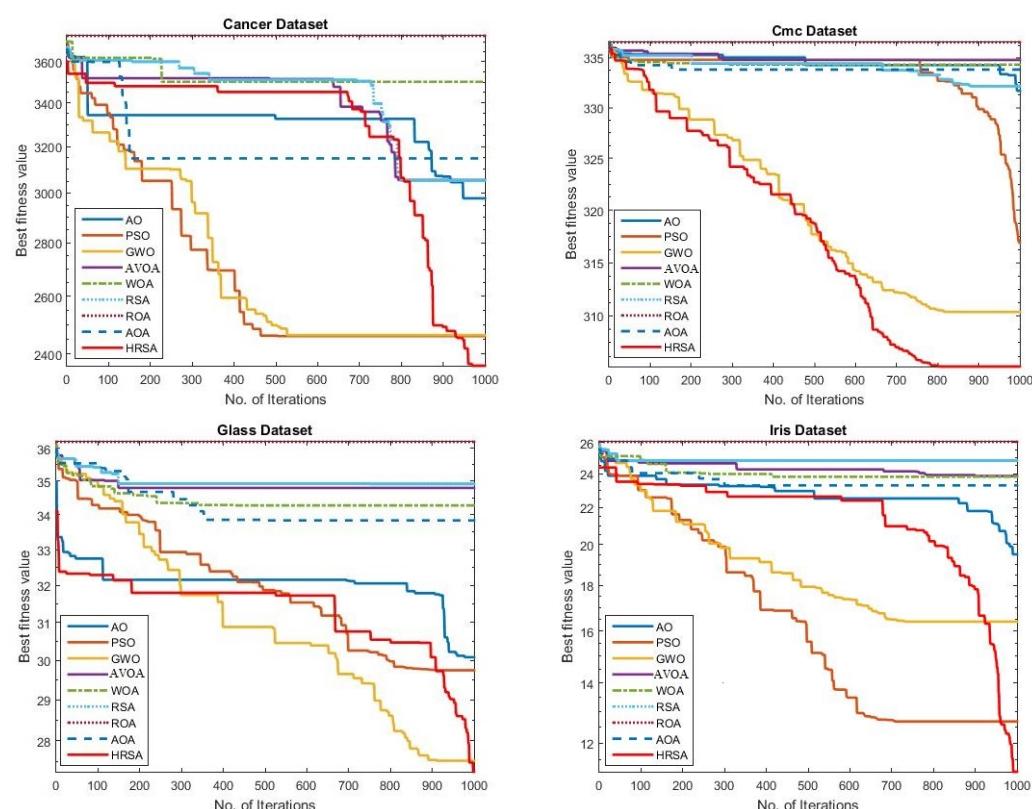


Figure 6. Cont.

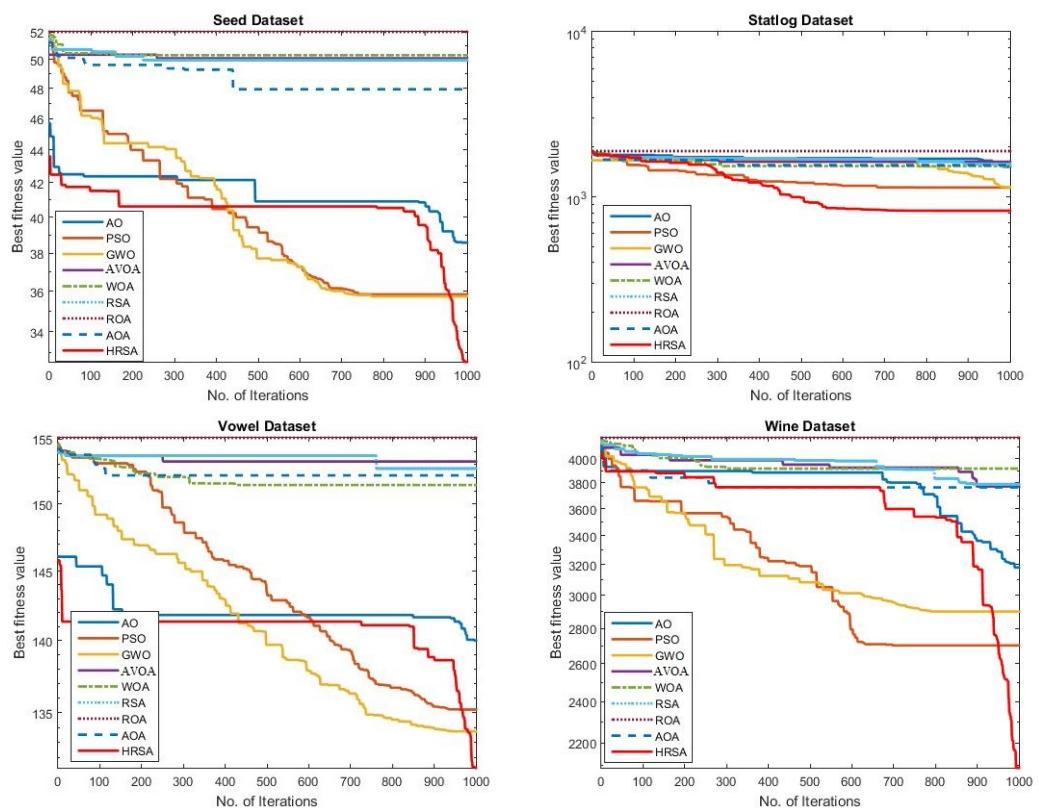


Figure 6. Convergence behaviour of the comparative algorithms using the tested data clustering problems.

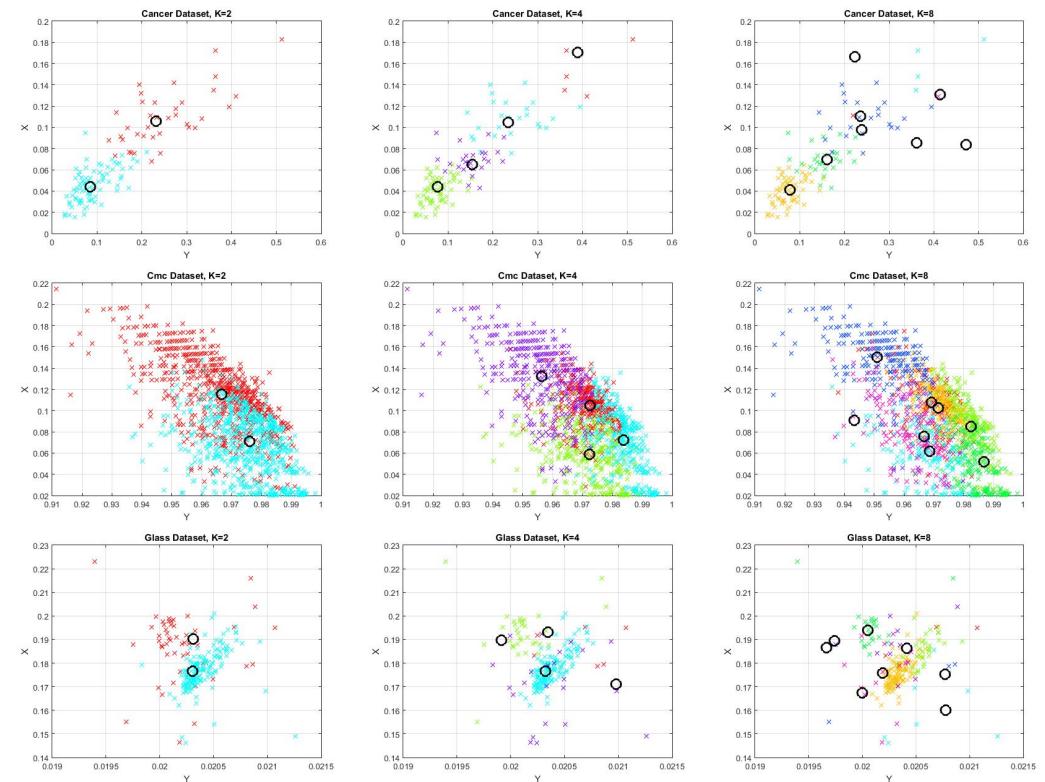


Figure 7. Cont.

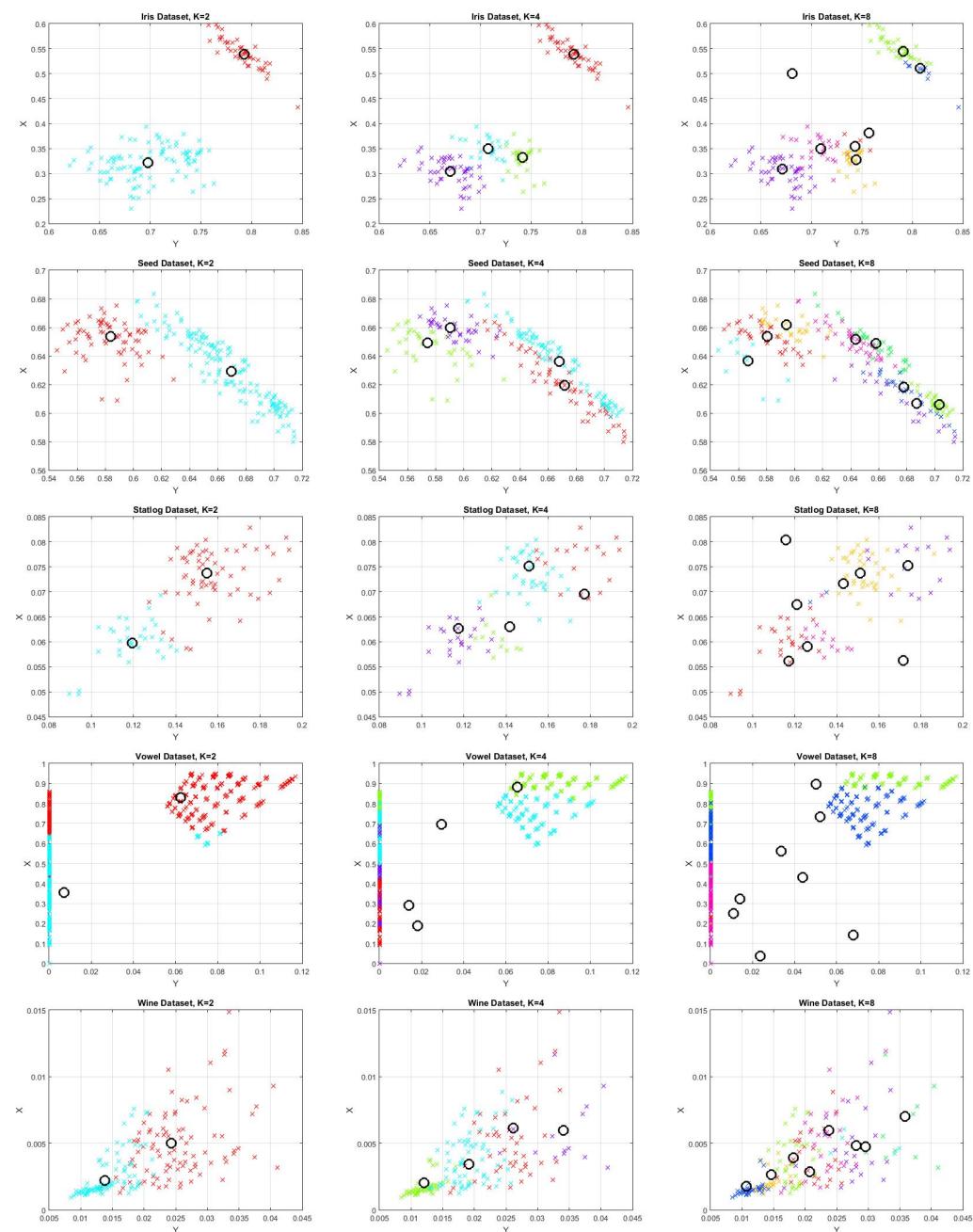


Figure 7. Clustering plot images using 8 datasets.

We concluded that the proposed method obtained promising results in almost all the tested problems. In comparison with other methods, the proposed method proved superior in solving the tested problems. As mentioned above, the proposed HRSA has been tested on solving the most common and standard benchmark functions; these functions are typically used to evaluate the performance of the optimization methods. According to the reported results, the proposed method obviously proved its ability to deal with different benchmark functions. Moreover, the data clustering problems are also used to evaluate the performance of the proposed method as a real-world application. The results given above showed that the proposed method obtained better results in almost all the tested problems and overwhelmed the comparative methods.



Figure 8. Execution time ranking for the data clustering problems.

4. Conclusions

Various optimization methods have been employed in the literature to solve different machine learning problems, especially clustering problems. Data clustering is one of the complex data mining problems that clusters the given objectives, which usually amounts to a massive number of data objects, into a predefined number of clusters. Optimization methods usually suffer from optimal local problems and unbalance between the search mechanisms (i.e., exploration and exploitation). This paper proposed a novel hybrid optimization search method for solving challenging optimization problems. The proposed method is called HRSA, which combined the original Reptile Search Algorithm (RSA) and Remora Optimization Algorithm (ROA) and handled these mechanisms' search process by a novel transition method. The proposed HRSA method solves the main weaknesses raised by the original methods and finds better solutions. The proposed HRSA is tested on various complicated optimization problems: twenty-three benchmark test functions and eight data clustering problems. The results illustrated that the proposed HRSA method performs significantly better than the original and comparative state-of-the-art methods. The proposed method was overwhelmingly better than all the other methods according to the mathematical problems. It obtained promising results in solving the clustering problems. Thus, HRSA has a remarkable efficacy when employed for various clustering problems.

In future work, the proposed method can be further studied to discover the weaknesses that can be improved. Additionally, it can be hybridized with other metaheuristic components. Moreover, the proposed HRSA can be investigated to solve other problems such as text clustering, text classification, image enhancement, image segmentation, task scheduling in computing, parameter estimation, forecasting problems, advanced mathematical problems, prediction problems, industrial problems, engineering problems, constrained mechanical design problems, home energy management, wastewater quality parameters, and other real-world problems [38–40]. The limitation of this paper is that the used data sets for the data clustering can be real data in the future. Additionally, in some cases, the proposed method needs more execution time.

Author Contributions: K.H.A.: Conceptualization, supervision, methodology, formal analysis, resources, data curation, and writing—original draft preparation. L.A.: Conceptualization, supervision, writing—review and editing, and project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is available from the authors upon reasonable request.

Acknowledgments: The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4320277DSR02).

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

References

1. Fiorini, L.; Cavallo, F.; Dario, P.; Eavis, A.; Caleb-Solly, P. Unsupervised machine learning for developing personalised behaviour models using activity data. *Sensors* **2017**, *17*, 1034. [CrossRef] [PubMed]
2. Alswaitti, M.; Albughdadi, M.; Isa, N.A.M. Density-based particle swarm optimization algorithm for data clustering. *Expert Syst. Appl.* **2018**, *91*, 170–186. [CrossRef]
3. Gao, K.; Khan, H.A.; Qu, W. Clustering with Missing Features: A Density-Based Approach. *Symmetry* **2022**, *14*, 60. [CrossRef]
4. Jain, A.K.; Murty, M.N.; Flynn, P.J. Data clustering: A review. *ACM Comput. Surv. (CSUR)* **1999**, *31*, 264–323. [CrossRef]
5. Kushwaha, N.; Pant, M.; Kant, S.; Jain, V.K. Magnetic optimization algorithm for data clustering. *Pattern Recognit. Lett.* **2018**, *115*, 59–65. [CrossRef]
6. Kant, S.; Ansari, I.A. An improved K means clustering with Atkinson index to classify liver patient dataset. *Int. J. Syst. Assur. Eng. Manag.* **2016**, *7*, 222–228. [CrossRef]
7. Rout, R.; Parida, P.; Alotaibi, Y.; Alghamdi, S.; Khalaf, O.I. Skin lesion extraction using multiscale morphological local variance reconstruction based watershed transform and fast fuzzy C-means clustering. *Symmetry* **2021**, *13*, 2085. [CrossRef]
8. Chander, S.; Vijaya, P.; Dhyani, P. Multi kernel and dynamic fractional lion optimization algorithm for data clustering. *Alex. Eng. J.* **2018**, *57*, 267–276. [CrossRef]
9. George, G.; Parthiban, L. Multi objective hybridized firefly algorithm with group search optimization for data clustering. In Proceedings of the 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), Kolkata, India, 20–22 November 2015; pp. 125–130.
10. Kumar, Y.; Kaur, A. Variants of bat algorithm for solving partitional clustering problems. *Eng. Comput.* **2021**, 1–27. Available online: <https://www.springerprofessional.de/en/variants-of-bat-algorithm-for-solving-partitional-clustering-pro/18956468> (accessed on 20 January 2022).
11. Mittal, H.; Pandey, A.C.; Saraswat, M.; Kumar, S.; Pal, R.; Modwel, G. A comprehensive survey of image segmentation: Clustering methods, performance parameters, and benchmark datasets. *Multimed. Tools Appl.* **2021**, 1–26. Available online: <https://link.springer.com/article/10.1007/s11042-021-10594-9> (accessed on 20 January 2022).
12. Yin, X.; Chen, S.; Hu, E.; Zhang, D. Semi-supervised clustering with metric learning: An adaptive kernel method. *Pattern Recognit.* **2010**, *43*, 1320–1333. [CrossRef]
13. Koryshev, N.; Hodashinsky, I.; Shelupanov, A. Building a fuzzy classifier based on whale optimization algorithm to detect network intrusions. *Symmetry* **2021**, *13*, 1211. [CrossRef]
14. Hussein, A.M.; Abdullah, R.; AbdulRashid, N. Flower Pollination Algorithm With Profile Technique For Multiple Sequence Alignment. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; pp. 571–576.
15. Hussein, A.M.; Abdullah, R.; AbdulRashid, N.; Ali, A.N.B. Protein multiple sequence alignment by basic flower pollination algorithm. In Proceedings of the 2017 8th International Conference on Information Technology (ICIT), Amman, Jordan, 17–18 May 2017; pp. 833–838.
16. Das, P.; Das, D.K.; Dey, S. A new class topper optimization algorithm with an application to data clustering. *IEEE Trans. Emerg. Top. Comput.* **2018**, *8*, 948–959. [CrossRef]
17. Alam, S.; Dobbie, G.; Riddle, P. An evolutionary particle swarm optimization algorithm for data clustering. In Proceedings of the 2008 IEEE Swarm Intelligence Symposium, St. Louis, MO, USA, 21–23 September 2008; pp. 1–6.
18. Agbaje, M.B.; Ezugwu, A.E.; Els, R. Automatic data clustering using hybrid firefly particle swarm optimization algorithm. *IEEE Access* **2019**, *7*, 184963–184984. [CrossRef]
19. Rahnema, N.; Gharehchopogh, F.S. An improved artificial bee colony algorithm based on whale optimization algorithm for data clustering. *Multimed. Tools Appl.* **2020**, *79*, 32169–32194. [CrossRef]
20. Bijari, K.; Zare, H.; Veisi, H.; Bobarshad, H. Memory-enriched big bang–big crunch optimization algorithm for data clustering. *Neural Comput. Appl.* **2018**, *29*, 111–121. [CrossRef]
21. Kao, Y.T.; Zahara, E.; Kao, I.W. A hybridized approach to data clustering. *Expert Syst. Appl.* **2008**, *34*, 1754–1762. [CrossRef]
22. Mahmoudi, M.R.; Akbarzadeh, H.; Parvin, H.; Nejatian, S.; Rezaie, V.; Alinejad-Rokny, H. Consensus function based on cluster-wise two level clustering. *Artif. Intell. Rev.* **2021**, *54*, 639–665. [CrossRef]
23. Mahapatra, S.; Dey, B.; Raj, S. A novel ameliorated Harris hawk optimizer for solving complex engineering optimization problems. *Int. J. Intell. Syst.* **2021**, *36*, 7641–7681. [CrossRef]

24. Raj, S.; Bhattacharyya, B. Optimal placement of TCSC and SVC for reactive power planning using Whale optimization algorithm. *Swarm Evol. Comput.* **2018**, *40*, 131–143. [[CrossRef](#)]
25. Shaikh, M.S.; Hua, C.; Raj, S.; Kumar, S.; Hassan, M.; Ansari, M.M.; Jatoi, M.A. Optimal parameter estimation of 1-phase and 3-phase transmission line for various bundle conductor's using modified whale optimization algorithm. *Int. J. Electr. Power Energy Syst.* **2022**, *138*, 107893. [[CrossRef](#)]
26. Shekarappa, G.S.; Mahapatra, S.; Raj, S. Voltage constrained reactive power planning problem for reactive loading variation using hybrid harris hawk particle swarm optimizer. *Electr. Power Components Syst.* **2021**, *49*, 421–435. [[CrossRef](#)]
27. Abualigah, L.; Abd Elaziz, M.; Sumari, P.; Geem, Z.W.; Gandomi, A.H. Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Syst. Appl.* **2022**, *191*, 116158. [[CrossRef](#)]
28. Jia, H.; Peng, X.; Lang, C. Remora optimization algorithm. *Expert Syst. Appl.* **2021**, *185*, 115665. [[CrossRef](#)]
29. Shinawi, A.E.; Ibrahim, R.A.; Abualigah, L.; Zelenakova, M.; Elaziz, M.A. Enhanced Adaptive Neuro-Fuzzy Inference System Using Reptile Search Algorithm for Relating Swelling Potentiality Using Index Geotechnical Properties: A Case Study at El Sherouk City, Egypt. *Mathematics* **2021**, *9*, 3295. [[CrossRef](#)]
30. Agushaka, J.O.; Ezugwu, A.E.; Abualigah, L. Dwarf Mongoose Optimization Algorithm. *Comput. Methods Appl. Mech. Eng.* **2022**, *391*, 114570. [[CrossRef](#)]
31. Oyelade, O.N.; Ezugwu, A.E.; Mohamed, T.I.; Abualigah, L. Ebola Optimization Search Algorithm: A new nature-inspired metaheuristic algorithm *IEEE Access* **2022**. [[CrossRef](#)]
32. Abualigah, L.; Yoursri, 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](#)]
33. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
34. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
35. Abdollahzadeh, B.; Gharehchopogh, F.S.; Mirjalili, S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Comput. Ind. Eng.* **2021**, *158*, 107408. [[CrossRef](#)]
36. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
37. Abualigah, L.; Diabat, A.; Mirjalili, S.; Abd Elaziz, M.; Gandomi, A.H. The arithmetic optimization algorithm. *Comput. Methods Appl. Mech. Eng.* **2021**, *376*, 113609. [[CrossRef](#)]
38. Gupta, S.; Abderazek, H.; Yıldız, B.S.; Yıldız, A.R.; Mirjalili, S.; Sait, S.M. Comparison of metaheuristic optimization algorithms for solving constrained mechanical design optimization problems. *Expert Syst. Appl.* **2021**, *183*, 115351. [[CrossRef](#)]
39. Wang, X.; Mao, X.; Khodaei, H. A multi-objective home energy management system based on internet of things and optimization algorithms. *J. Build. Eng.* **2021**, *33*, 101603. [[CrossRef](#)]
40. Abunama, T.; Ansari, M.; Awolusi, O.O.; Gani, K.M.; Kumari, S.; Bux, F. Fuzzy inference optimization algorithms for enhancing the modelling accuracy of wastewater quality parameters. *J. Environ. Manag.* **2021**, *293*, 112862. [[CrossRef](#)] [[PubMed](#)]