

6-1-2021

Niching grey wolf optimizer for multimodal optimization problems

Rasel Ahmed
Universiti Teknologi Petronas

Amril Nazir
Zayed University

Shuhaimi Mahadzir
Universiti Teknologi Petronas

Mohammad Shorfuzzaman
Taif University

Jahedul Islam
Universiti Teknologi Petronas

Follow this and additional works at: <https://zuscholars.zu.ac.ae/works>



Part of the [Computer Sciences Commons](#)





Recommended Citation

Ahmed, Rasel; Nazir, Amril; Mahadzir, Shuhaimi; Shorfuzzaman, Mohammad; and Islam, Jahedul, "Niching grey wolf optimizer for multimodal optimization problems" (2021). *All Works*. 4284.
<https://zuscholars.zu.ac.ae/works/4284>

This Article is brought to you for free and open access by ZU Scholars. It has been accepted for inclusion in All Works by an authorized administrator of ZU Scholars. For more information, please contact Yrjo.Lappalainen@zu.ac.ae, nikesh.narayanan@zu.ac.ae.

Article

Niching Grey Wolf Optimizer for Multimodal Optimization Problems

Rasel Ahmed ¹, Amril Nazir ^{2,*}, Shuhaimi Mahadzir ^{1,3}, Mohammad Shorfuzzaman ⁴ and Jahedul Islam ⁵

¹ Department of Chemical Engineering, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Perak Darul Ridzuan, Malaysia; rasel_19001045@utp.edu.my (R.A.); shuham@utp.edu.my (S.M.)

² Department of Information Systems, College of Technological Innovation, Abu Dhabi Campus, Zayed University, Abu Dhabi P.O. Box 144534, United Arab Emirates

³ Centre for Process Systems Engineering, Institute of Autonomous System, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Perak Darul Ridzuan, Malaysia

⁴ Department of Computer Science, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia; m.shorf@tu.edu.sa

⁵ Department of Fundamental and Applied Sciences, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Perak Darul Ridzuan, Malaysia; jahedul_17010697@utp.edu.my

* Correspondence: mohd.nazir@zu.ac.ae

Abstract: Metaheuristic algorithms are widely used for optimization in both research and the industrial community for simplicity, flexibility, and robustness. However, multi-modal optimization is a difficult task, even for metaheuristic algorithms. Two important issues that need to be handled for solving multi-modal problems are (a) to categorize multiple local/global optima and (b) to uphold these optima till the ending. Besides, a robust local search ability is also a prerequisite to reach the exact global optima. Grey Wolf Optimizer (GWO) is a recently developed nature-inspired metaheuristic algorithm that requires less parameter tuning. However, the GWO suffers from premature convergence and fails to maintain the balance between exploration and exploitation for solving multi-modal problems. This study proposes a niching GWO (NGWO) that incorporates personal best features of PSO and a local search technique to address these issues. The proposed algorithm has been tested for 23 benchmark functions and three engineering cases. The NGWO outperformed all other considered algorithms in most of the test functions compared to state-of-the-art metaheuristics such as PSO, GSA, GWO, Jaya and two improved variants of GWO, and niching CSA. Statistical analysis and Friedman tests have been conducted to compare the performance of these algorithms thoroughly.

Keywords: metaheuristic algorithm; swarm intelligence; multi-modal optimization; Grey Wolf Optimizer; niching technique; local search



Citation: Ahmed, R.; Nazir, A.; Mahadzir, S.; Shorfuzzaman, M.; Islam, J. Niching Grey Wolf Optimizer for Multimodal Optimization Problems. *Appl. Sci.* **2021**, *11*, 4795. <https://doi.org/10.3390/app11114795>

Received: 8 April 2021
Accepted: 4 May 2021
Published: 24 May 2021

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



Copyright: © 2021 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

The area of nature-inspired metaheuristic algorithms is continuously evolving with newly developed algorithms. These algorithms are well known for their widespread local and global search ability, local optima avoidance, and quick convergence ability. In addition, these algorithms do not need any knowledge of a function's gradient or differentiability [1,2]. Since the metaheuristic algorithms have these superior properties with easy applicability and fewer parameter requirements, significant variants of these algorithms are developed [3]. The most widely used algorithms are Simulated Annealing (SA) [4], Genetic Algorithm (GA) [5], Particle Swarm Optimization (PSO) [6], Differential Evolution (DE) [7], and Ant Colony Optimization (ACO) [8]. A few recent metaheuristics that seek the attention of researchers is Bee Collecting Pollen Algorithm (BCPA) [9], Black Hole (BH) [10] Algorithm, Artificial Bee Colony Algorithm (ABC) [11], Central Force Optimization (CFO) [10], Electro-search Algorithm (ESA) [12], Gravitational Search Algorithm

(GSA) [13], Teaching Learning Based Algorithm (TLBO) [14], Artificial Chemical Reaction Optimization Algorithm (ACROA) [15], Firefly Algorithm (FA) [16], Big Bang–Big Crunch (BBBC) [17], Grey Wolf Optimizer (GWO) [18], Dragon Fly Algorithm [19], and Harris Hawk Optimizer (HHO) [20], Jaya Algorithm [21]. Among these recent metaheuristics, the GSA, GWO, and Jaya Algorithm is well known for simplicity, effective exploration and exploitation.

Among the algorithms mentioned above, the GWO is a novel metaheuristic algorithm that falls under swarm intelligence [18]. This algorithm has been proposed mimicking the organizational order of wolves and their collective hunting nature. Due to its simplicity, fast seeking speed, good precision for search, and fewer controlling parameters, this algorithm is successfully applied to various technical fields such as feature selection [22–24], wind speed forecasting [25], optimal power flow [26,27], pattern recognition [28], unit commitment [29,30] parameters estimation [31,32], wellbore trajectory optimization [31]. In addition to these applications, the GWO and variants also performed well for combinatorial optimization problems and successfully applied to different fields, such as vehicle routing problem [32], job scheduling problems [33–35], and economic dispatch [36,37] problems. Some modifications to the GWO algorithm and its application in various optimization fields are now described.

Quasi-oppositional based learning (Q-OBL) GWO called QOGWO was proposed by Guha et al. [38] to control power systems load frequency. The authors designed optimal PID controller by using QOGWO for two areas and four hydrothermal power plants and showed that their proposed methods achieved promising performance like fuzzy logic, ANN, and ANFIS system. In a different research, crossover and mutation-based hybrid GWO was proposed by Jayabarathi et al. [36] to optimize non-linear, non-convex, constrained, and unconstrained economic dispatch problems at different conditions. The authors claimed that their method showed promising performance. Korayem et al. hybridized the GWO with K-means clustering technique to solve CVRP (capacitated vehicle routing problems) and named it as K-GWO [32]. The authors further integrated a capacity constraint to their algorithm and proposed 2 clustering techniques. They follow the cluster-first route-second method to solve CVRP problems and tested their proposed technique for 20 benchmarks, and the obtained solutions are within a mean deviation of 1.76% from the known optimum solutions. Singh and Singh [39] proposed a hybridization of GWO with SCA in 2017. The hybrid GWOSCA shows good performance for unimodal functions but failed to perform better for multi-modal problems [39]. Jiang and Zhang applied a hybrid Grey Wolf Optimizer algorithm for job shop scheduling and flexible job shop scheduling problems. Their hybrid algorithm is incorporated with crossover operator to handle discrete variables, whereas mutation operator helps to avoid local optima, and a neighborhood search is applied to increase exploration. Their proposed algorithm outperforms PAGA, LSGA, TLBO, EDA, BFO algorithms for both case studies [40]. Long et al. [41] recommended exploration-enhanced GWO (EEGWO), in which they introduced a new formula for updating the position by selecting a random wolf to guide the search and non-linearly controlling the GWO parameter “a”. Even after making these modifications, their proposed algorithm still suffers from premature convergence and local optima. Dhargupta et al. [42] designed the selective opposition-based learning GWO (SOGWO) to improve GWO’s convergence ability. To govern the omega wolves, they used Spearman’s correlation coefficient and chose a few dimensions of wolves to apply opposition, which enhanced the exploration ability of the algorithm. A recent variation of the GWO is developed by Shahraki et al. [43], where they suggested a new movement technique encouraged by the discrete hunting nature of wolves called dimension learning-based hunting (DLH) search. According to the DLH, all wolves create their neighbors, sharing information among themselves.

The literature mentioned above indicates that due to easy implementation and less parameter tuning, GWO variants are already applied in every optimization field and have shown good performances empirically. However, premature convergence and the inability to balance the exploitation with exploration are still the main drawbacks of the GWO and

its variants for solving complex multi-modal optimization problems [42,43]. To overcome these limitations, we make a decent attempt to improve the GWO algorithm by adapting the niching technique. Niching techniques are well known to trace multiple optimal solutions together. These techniques increase population diversity and reduce the likelihood of being trapped in local solutions. In one iteration, they are also able to trace numerous local/global optima. A few standard niching techniques developed in recent years are clustering [44], fitness sharing [44], crowding [45], and clearing [46]. Nevertheless, niching methods need more function evaluations to reach the global optima. It has been seen from literature that various niching methods are applied to different types of metaheuristic algorithms such as GA [44–46], PSO [2,47–49], CSA [50], and niching technique has significantly improved the performance of these optimization algorithms. Inspired by these results, we incorporate a fitness Euclidean distance ratio (FER) based niching technique with modified velocity update and local search into the GWO algorithm, which has shown to significantly improve the basic GWO algorithm's accuracy. Our proposed method also does not need any tuning parameters, which is a significant advantage. The research's key contribution is to (a) integrate FER niching technique into the GWO algorithm, (b) modify the velocity update equation, and (c) apply a local search technique at the end of the search process. To the best of our knowledge, this study first incorporates the niching technique to the GWO algorithm. Since the multi-modal function optimization also needs to allocate the desired global optima more precisely, we further incorporated the personal best feature of PSO to hold the good solutions. The remaining part of the article is organized as follows: Section 2 describes the NGWO algorithm. Section 3 illustrates the results and engineering applications of NGWO. Finally, Section 4 presents the conclusion.

2. Materials and Methods

2.1. Grey Wolf Optimizer

Mirjalili et al. [18] suggested GWO, a novel population-based algorithm, which falls under the swarm intelligence category. The key feature of GWO that makes it less prone to local solutions and better than other algorithms is that it searches for the best solutions by following three leader wolves: alpha, beta, and delta, while most other algorithms only follow one best solution. The common wolves are called omega, and they always follow the three leaders. The major steps of GWO are:

- Tracking, chasing, proceeding towards the target;
- Chasing, encompassing, distressing the target;
- Attacking the target.

The encircling mechanism can be mathematically represented as:

$$\vec{X}_{(k+1)} = \vec{X}_{P(k)} - \vec{A} \cdot |\vec{C} \cdot \vec{X}_{P(k)} - \vec{X}_{(k)}| \quad (1)$$

Here, vector \vec{X} indicates the position of wolves, \vec{X}_p represents optimal solution's position, \vec{C} and \vec{A} are acceleration coefficients, $\vec{C} = 2 \cdot \vec{r}_2$ and $\vec{A} = 2 \vec{a} \cdot \vec{r}_1 - \vec{a}$, where \vec{r}_1 , and \vec{r}_2 are randomly generated vectors within [0,1]. \vec{a} is a constant vector that declines linearly from 2 to 0. All the wolves update their positions by following these equations.

$$\vec{X}_{(a)} = \vec{X}_{\alpha} - \vec{A} \cdot |\vec{C}_1 \cdot \vec{X}_{\alpha} - \vec{X}| \quad (2)$$

$$\vec{X}_{(b)} = \vec{X}_{\beta} - \vec{A} \cdot |\vec{C}_2 \cdot \vec{X}_{\beta} - \vec{X}| \quad (3)$$

$$\vec{X}_{(c)} = \vec{X}_{\delta} - \vec{A} \cdot |\vec{C}_3 \cdot \vec{X}_{\delta} - \vec{X}| \quad (4)$$

$$\vec{X} = \frac{(\vec{X}_{(a)} + \vec{X}_{(b)} + \vec{X}_{(c)})}{3} \quad (5)$$

At each iteration, the wolves will update their positions according to Equations (1)–(5). The parameter \vec{a} controls the movement of the wolves during the first half of the iteration period where $\vec{a} > 1$, and it will intensify the exploration ability of the algorithm. In the last half of the iteration period where $\vec{a} < 1$, it will increase the wolves' exploitation ability.

2.2. Niching Technique

Niching techniques have the superiority of containing multiple solutions, which help them to handle the multi-modal functions and effectively reach the global optimal basin. In the proposed technique, the fitness Euclidean-distance ratio GWO (FER-GWO) is employed to locate the best closest solutions, and the FER-GWO is developed following the concept of FDR-PSO [51]. To calculate the FER value between two wolves i and j , the following equation has been utilized.

$$\text{FER}_{ij} = (\alpha) \cdot \frac{F(p_i) - F(p_j)}{\|p_i - p_j\|} \quad (6)$$

Here, p_i and p_j are the positions of the corresponding i and j wolves, and $F(p_i)$ and $F(p_j)$ represent their cost function values.

$$\text{The scaling factor } (\alpha) = \frac{\|S\|}{F(p_b) - F(p_w)} \quad (7)$$

In the above equation $F(p_w)$ and $F(p_b)$ are the cost function values of the worst and best wolves among the current population, and the search space size is calculated based on decision variables bounds, and numerically represented as:

$$S = \sqrt{\sum_{n=1}^D (x_n^u - x_n^l)^2} \quad (8)$$

Since, multi-modal optimization needs extensive exploration, niching is an efficient technique that could meet the requirement. Therefore, in the proposed GWO algorithm, we incorporated the personal best features to keep track of the best solutions discovered so far during the search process. The wolves will be guided towards more fittest neighborhood points at each iteration that can be obtained by calculating the FER values. The FER value has superiority in that it does not need parameters specification. Moreover, bigger population size is recommended for niching to locate all the local and global optima.

2.3. Modified Velocity Update Equation

During the iteration period, we determine the $nbest_i$ of each wolf based on the FER value from Equation (6). The $nbest_i$ value is further used to modify the velocity update Equation of beta and delta wolves, which is formulated as:

$$\vec{X}_{(b)} = nbest_i - \vec{A} \cdot |\vec{C} \cdot nbest_i - \vec{X}| \quad (9)$$

$$\vec{X}_{(c)} = nbest_i - \vec{A} \cdot |\vec{C} \cdot nbest_i - \vec{X}| \quad (10)$$

We set a niching constant (NC) as 0.5 and generate a random number (r_i) during each iteration, if $r_i > NC$, then the beta and delta wolves will be guided by $nbest_i$ using Equations (9) and (10). Otherwise, the original Equations (3) and (4) will be used to guide the beta and delta wolves. Please note that the alpha wolf is always guided by the original position update equation (Equation (2)).

2.4. Local Search

It is challenging for the niching techniques to hit the global optima accurately in complex multi-modal problems reliably. We incorporated a local search technique into our NGWO to generate offspring close to the personal best (Pbest). It also improves the algorithm's fine search capability, which increases the likelihood of obtaining optimal results. The definition of Qu et al. [2] is used to establish the local search technique, and the pseudo-code is provided below in Algorithm 1.

Algorithm 1 Local Search

```

1: Update the present Pbest by NGWO
2: for  $i = 1$  to NP (number of wolves)
3:   Search  $Pbest\_nearesti$  (the nearest  $Pbest$  member to  $Pbesti$ )
4:   if  $fit(Pbest\_nearesti) <= fit(Pbesti)$ 
5:      $Temp = Pbesti + 1.5 * rand * (Pbest\_nearesti - Pbesti)$ 
6:   else
7:      $Temp = Pbesti + 1.5 * rand * (Pbesti - Pbest\_nearesti)$ 
8:   end if
9:   Check for bounds violation, and assess Temp.
10:  if  $fit(Temp) < fit(Pbesti)$ 
11:     $Pbesti = Temp$ 
12:  end if
13: end for

```

During each iteration of the NGWO, the niching technique is applied by using Equations (6)–(8). The wolves update their positions based on Equations (2)–(5) and (9)–(10). Finally, local search is applied to the wolves' personal best (Pbest). Figure 1 shows a detailed flow map of the proposed algorithm.

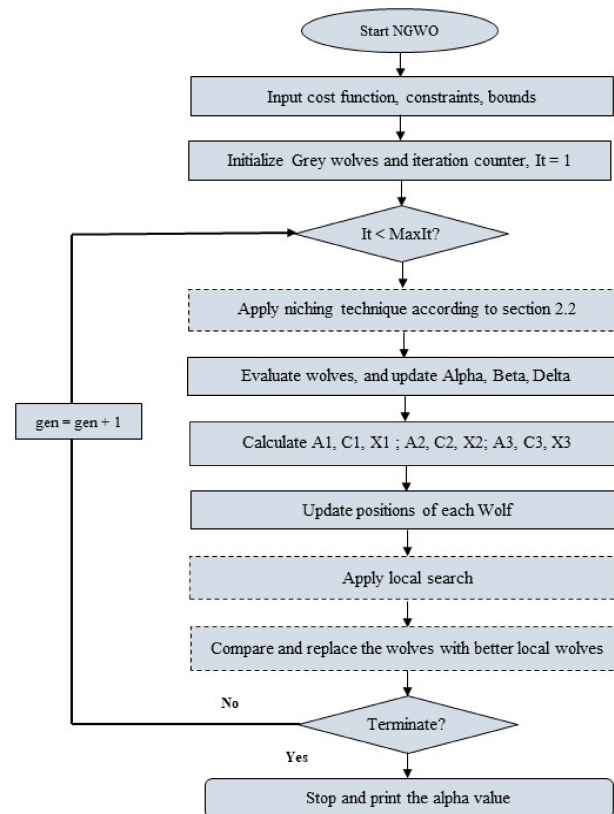


Figure 1. Steps of the NGWO algorithm.

3. Result and Discussion

3.1. Experimental Setup

We evaluated the proposed algorithm's performance comprehensively for 23 test functions as a minimization problem [13,18]. These functions are very common for testing any algorithm's effectiveness, consisting of 7 unimodal, 6 multi-modal, and 10 fixed dimensional multi-modal functions. The population and iteration number are set as 50 and 1000 for the proposed method, respectively. We keep these two features similar to the other compared studies to make the comparison fair and non-biased. The compared PSO, GSA, GWO, and SOGWO values are taken from [42], and niching CSA values are inserted from [50]. The Matlab code of Jaya algorithm and IGWO algorithms are collected from the authors website for comparison. The test functions are numerically presented here, and their detailed properties are described in Table 1.

$$F_1(y) = \sum_{j=1}^D y_j^2 \quad (11)$$

$$F_2(y) = \sum_{j=1}^D |y_j| + \prod_{j=1}^D |y_j| \quad (12)$$

$$F_3(y) = \sum_{j=1}^D \sum_{i=1}^i y_j^2 \quad (13)$$

$$F_4(y) = \{|y_j|, 1 \leq j \leq n\} \quad (14)$$

$$F_5(y) = \sum_{j=1}^{D-1} \left[100 \times (y_j^2 - y_{j+1})^2 + (y_j - 1)^2 \right] \quad (15)$$

$$F_6(y) = \sum_{j=1}^D (|y_j + 0.5|)^2 \quad (16)$$

$$F_7(y) = \sum_{j=1}^D j y_j^4 + \text{random}[0, 1] \quad (17)$$

$$F_8(y) = \sum_{j=1}^D -y_j \sin\left(\sqrt{|y_j|}\right) \quad (18)$$

$$F_9(y) = \sum_{j=1}^D (y_j^2 - 10 \times \cos(2\pi y_j) + 10) \quad (19)$$

$$F_{10}(y) = -20 \times \exp \exp \left(-0.2 \times \sqrt{\frac{1}{D} \sum_{j=1}^D y_j^2} \right) - \exp \exp \left(\frac{1}{D} \sqrt{\sum_{j=1}^D \cos 2\pi y_j} \right) + 20 + e \quad (20)$$

$$F_{11}(y) = \sum_{j=1}^D \frac{y_j^2}{4000} - \prod_{j=1}^D \cos\left(\frac{y_j}{\sqrt{j}}\right) + 1 \quad (21)$$

$$F_{12}(y) = \frac{\pi}{n}(\pi x) + \sum_{j=1}^D (y_j - 1)^2 [1 + 10(\pi x_{j+1})] + (x_n - 1)^2 + \sum_{j=1}^D u(y_j, 10, 100, 4) \quad (22)$$

$$x_j = 1 + \frac{y_j + 1}{4}, u(y_j, a, k, m) = k(y_j - a), \text{ if } y_j > a; \text{ or } 0, \text{ if } -a < y_j < a; \text{ or } k(-y_j - a)^m, \text{ if } y_j < a \quad (23)$$

$$F_{13}(y) = (3\pi y_j) + \sum_{j=1}^D (y_j - 1)^2 [1 + (3\pi y_j + 1)] + (y_n - 1)^2 [1 + (2\pi y_n)] + \sum_{j=1}^D u(y_j, 5, 100, 4)$$

$$F_{14}(y) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{i + \sum_{i=1}^2 (y_i - a_{i,j})} \right)^{-1} \tag{24}$$

$$F_{15}(y) = \sum_{i=1}^{11} \left[a_i - \frac{y_i (b_i^2 + b_i y_2)}{b_i^2 + b_i y_3 + y_4} \right]^2 \tag{25}$$

$$F_{16}(y) = 4y_1^2 - 2.1y_1^4 + \left(\frac{1}{3} \right) y_1^6 + y_1 y_2 - 4y_2^2 + 4y_2^4 \tag{26}$$

$$F_{17}(y) = (y_2 - \left(\frac{5.1}{4\pi^2} \right) y_1^2 + \left(\frac{5}{\pi} \right) y_1 - 6) + 10 (1 - (1/8\pi)) \cos y_1 + 10 \tag{27}$$

$$F_{18}(y) = \left[1 + (y_1 + y_2 + 1)^2 (19 - 14y_1 + 3y_1^2 - 14y_2 + 6y_1 y_2 + 3y_2^2) \right] \times \left[30 + (2y_1 - 3y_2)^2 \times (18 - 32y_1 + 12y_1^2 + 48y_2 - 36y_1 y_2 + 27y_2^2) \right] \tag{28}$$

$$F_{19}(y) = - \sum_{i=1}^4 c_i \exp(- \sum_{j=1}^3 a_{i,j} (y_j - p_{i,j})^2) \tag{29}$$

$$F_{20}(y) = - \sum_{i=1}^4 c_i \exp(- \sum_{j=1}^6 a_{i,j} (y_j - p_{i,j})^2) \tag{30}$$

$$F_{21}(y) = - \sum_{i=1}^5 \left[(y - a_i)(y - a_i)^T + c_i \right]^{-1} \tag{31}$$

$$F_{22}(y) = - \sum_{i=1}^7 \left[(y - a_i)(y - a_i)^T + c_i \right]^{-1} \tag{32}$$

$$F_{23}(y) = - \sum_{i=1}^{10} \left[(y - a_i)(y - a_i)^T + c_i \right]^{-1} \tag{33}$$

Table 1. Benchmark function properties.

Functions	Dimension	Range	Global Optima
F ₁	30	[-100, 100]	0
F ₂	30	[-10, 10]	0
F ₃	30	[-100, 100]	0
F ₄	30	[-100, 100]	0
F ₅	30	[-30, 30]	0
F ₆	30	[-100, 100]	0
F ₇	30	[-1.28, 1.28]	0
F ₈	30	[-500, 500]	-418 × D
F ₉	30	[-5.12, 5.12]	0
F ₁₀	30	[-32, 32]	0
F ₁₁	30	[-600, 600]	0
F ₁₂	30	[-50, 50]	0
F ₁₃	30	[-50, 50]	0
F ₁₄	2	[-65, 65]	1
F ₁₅	4	[-5, 5]	0.0003
F ₁₆	2	[-5, 5]	-1.0316
F ₁₇	2	[-5, 5]	0.398
F ₁₈	2	[-2, 2]	3
F ₁₉	3	[0, 1]	-3.86
F ₂₀	6	[0, 10]	-3.32
F ₂₁	4	[0, 10]	-10.1532
F ₂₂	4	[0, 10]	-10.4028
F ₂₃	4	[0, 10]	-10.5363

Table 2 displays the parameter configurations of the compared algorithms. For each test function, we run the algorithms 30 times in MATLAB, and the performance indexes

(mean, standard deviation (SD), minimum (min) values) and Friedman test results are collected from the MATLAB and inserted in Tables 3–5, with the best values bolded.

Table 2. Parameters of the compared algorithms.

Algorithm	Parameters and Their Numerical Values
PSO	$c1 = 2, c1 = 2$
GSA	$c1 = 2, c2 = 2, G_0 = 1$
GWO	$a = 2$ to 0
JAYA	N/A
SOGWO	$a = 2$ to 0
IGWO	$a = 2$ to 0
NCSA	Awareness probability = 0.30, flight length = 0.20
NGWO	$a = 2$ to 0, NC = 0.5

Table 3. Results of unimodal benchmarks (F₁–F₇).

Func-tions	Index	PSO (1998)	GSA (2011)	GWO (2014)	JAYA (2016)	SOGWO (2020)	IGWO (2021)	NCSA (2020)	NGWO (This Work)
F ₁	Mean	1.30×10^{-11}	2.00×10^{-17}	1.36×10^{-70}	1.20×10^{-8}	6.05×10^{-77}	2.98×10^{-71}	3.48×10^{-91}	3.69×10^{-96}
	SD	8.80×10^{-11}	5.50×10^{-18}	2.57×10^{-70}	3.40×10^{-8}	1.49×10^{-76}	5.60×10^{-71}	1.05×10^{-90}	8.25×10^{-96}
	Min	1.10×10^{-15}	1.10×10^{-17}	9.07×10^{-73}	1.71×10^{-10}	3.81×10^{-79}	1.52×10^{-73}	N/A	1.38×10^{-107}
F ₂	Mean	2.90×10^{-6}	2.40×10^{-8}	5.64×10^{-41}	2.66×10^{-4}	1.18×10^{-44}	1.30×10^{-42}	6.64×10^{-61}	1.09×10^{-73}
	SD	1.30×10^{-5}	4.40×10^{-9}	6.43×10^{-41}	1.70×10^{-4}	1.34×10^{-44}	1.26×10^{-42}	1.71×10^{-60}	5.96×10^{-73}
	Min	4.40×10^{-9}	1.40×10^{-8}	3.59×10^{-42}	1.45×10^{-6}	3.50×10^{-44}	1.36×10^{-43}	N/A	1.46×10^{-88}
F ₃	Mean	1.20×10^2	2.30×10^2	1.09×10^{-19}	4.13×10^0	5.40×10^{-22}	3.29×10^{-14}	1.60×10^{-27}	1.29×10^{-9}
	SD	7.50×10^1	1.00×10^2	3.11×10^{-19}	8.26×10^0	2.60×10^{-21}	9.26×10^{-14}	5.97×10^{-27}	2.67×10^{-9}
	Min	1.90×10^1	7.50×10^1	2.56×10^{-25}	2.19×10^{-1}	1.17×10^{-28}	1.69×10^{-18}	N/A	1.70×10^{-18}
F ₄	Mean	4.20×10^{-1}	6.40×10^{-2}	1.94×10^{-17}	1.52×10^0	1.18×10^{-19}	1.01×10^{-14}	1.67×10^{-25}	4.53×10^{-32}
	SD	1.90×10^{-1}	2.50×10^{-1}	3.68×10^{-17}	1.04×10^0	1.51×10^{-19}	1.64×10^{-14}	7.64×10^{-25}	1.01×10^{-31}
	Min	1.40×10^{-1}	2.10×10^{-9}	1.28×10^{-18}	2.92×10^{-1}	7.08×10^{-21}	1.72×10^{-16}	N/A	8.49×10^{-40}
F ₅	Mean	2.70×10^1	2.80×10^1	2.63×10^1	3.73×10^1	2.65×10^1	2.74×10^1	2.68×10^1	2.48×10^1
	SD	8.40×10^0	1.00×10^1	6.69×10^{-1}	2.54×10^1	7.62×10^{-1}	3.05×10^{-1}	4.18×10^{-1}	1.86×10^{-1}
	Min	2.50×10^1	2.60×10^1	2.51×10^1	8.44×10^0	2.50×10^1	2.47×10^1	N/A	2.45×10^1
F ₆	Mean	1.30×10^{-12}	0.00×10^0	4.12×10^{-1}	1.21×10^{-8}	2.83×10^{-1}	1.00×10^{-5}	3.16×10^{-1}	2.66×10^{-4}
	SD	7.10×10^{-12}	0.00×10^0	2.45×10^{-1}	2.28×10^{-8}	2.47×10^{-1}	3.07×10^{-6}	2.39×10^{-1}	1.24×10^{-4}
	Min	8.30×10^{-16}	0.00×10^0	1.09×10^{-5}	1.24×10^{-10}	6.19×10^{-6}	4.87×10^{-6}	N/A	1.61×10^{-4}
F ₇	Mean	7.00×10^{-3}	2.80×10^{-2}	5.68×10^{-4}	2.87×10^{-2}	4.93×10^{-4}	7.60×10^{-4}	8.06×10^{-4}	3.41×10^{-4}
	SD	2.50×10^{-3}	1.70×10^{-2}	3.54×10^{-4}	1.09×10^{-2}	2.71×10^{-4}	2.94×10^{-4}	4.80×10^{-4}	2.26×10^{-4}
	Min	1.70×10^{-3}	8.40×10^{-3}	1.49×10^{-4}	1.27×10^{-2}	8.57×10^{-5}	3.33×10^{-4}	N/A	8.38×10^{-5}

Table 4. Results of the compared algorithms for multi-modal benchmarks (F₈–F₂₃).

Func-tions	Index	PSO (1998)	GSA (2011)	GWO (2014)	JAYA (2016)	SOGWO (2020)	IGWO (2021)	NCSA (2020)	NGWO (This Work)
F ₈	Mean	-9.00×10^3	-2.70×10^3	-6.07×10^3	-7.66×10^3	-6.57×10^3	-9.53×10^3	-7.25×10^3	-1.25×10^4
	SD	5.20×10^2	4.70×10^2	5.37×10^2	1.01×10^3	8.03×10^2	1.40×10^3	9.86×10^2	9.77×10^1
	Min	-1.00×10^4	-4.20×10^3	-7.09×10^3	-9.66×10^3	-8.18×10^3	-1.13×10^4	N/A	-1.26×10^4
F ₉	Mean	4.10×10^1	1.70×10^1	5.20×10^0	2.68×10^1	0.00×10^0	1.42×10^1	0.00×10^0	0.00×10^0
	SD	1.50×10^1	4.30×10^0	1.89×10^0	9.89×10^0	0.00×10^0	5.80×10^0	0.00×10^0	0.00×10^0
	Min	1.80×10^1	9.00×10^0	0.00×10^0	1.49×10^1	0.00×10^0	2.99×10^0	N/A	0.00×10^0
F ₁₀	Mean	9.10×10^{-8}	3.40×10^{-9}	1.31×10^{-14}	2.63×10^0	8.88×10^{-16}	9.41×10^{-15}	4.44×10^{-8}	4.44×10^{-15}
	SD	2.00×10^{-7}	4.10×10^{-10}	2.73×10^{-15}	9.95×10^{-1}	0.00×10^0	2.74×10^{-15}	0.00×10^0	0.00×10^0
	Min	4.60×10^{-9}	2.20×10^{-9}	7.99×10^{-15}	9.31×10^{-1}	8.88×10^{-16}	7.99×10^{-15}	N/A	4.44×10^{-15}
F ₁₁	Mean	1.20×10^{-2}	4.30×10^0	5.23×10^{-4}	1.99×10^{-2}	0.00×10^0	9.55×10^{-3}	0.00×10^0	0.00×10^0
	SD	1.20×10^{-2}	1.60×10^0	2.61×10^{-3}	2.97×10^{-2}	0.00×10^0	3.60×10^{-3}	0.00×10^0	0.00×10^0
	Min	5.10×10^{-15}	2.00×10^0	0.00×10^0	8.43×10^{-10}	0.00×10^0	0.00×10^0	N/A	0.00×10^0
F ₁₂	Mean	1.50×10^{-13}	2.50×10^{-2}	2.66×10^{-2}	1.58×10^{-1}	5.61×10^{-2}	7.50×10^{-7}	1.66×10^{-2}	2.10×10^{-5}
	SD	3.60×10^{-13}	6.10×10^{-2}	1.55×10^{-2}	2.26×10^{-1}	1.42×10^{-2}	1.81×10^{-7}	8.75×10^{-3}	3.62×10^{-6}
	Min	1.60×10^{-13}	6.20×10^{-2}	6.57×10^{-2}	4.84×10^{-11}	2.62×10^{-2}	4.45×10^{-7}	N/A	1.69×10^{-5}

Table 4. Cont.

Functions	Index	PSO (1998)	GSA (2011)	GWO (2014)	JAYA (2016)	SOGWO (2020)	IGWO (2021)	NCSA (2020)	NGWO (This Work)
F ₁₃	Mean	2.00×10^{-31}	2.10×10^{-18}	3.25×10^{-1}	1.97×10^0	3.53×10^{-1}	1.42×10^{-5}	4.08×10^{-1}	1.12×10^{-2}
	SD	4.30×10^{-31}	5.00×10^{-19}	1.56×10^{-1}	3.71×10^0	1.28×10^{-1}	4.31×10^{-6}	2.27×10^{-1}	1.07×10^{-2}
	Min	9.90×10^{-31}	1.22×10^{-18}	1.58×10^{-5}	6.58×10^{-9}	1.42×10^{-5}	7.42×10^{-6}	N/A	3.42×10^{-4}
F ₁₄	Mean	1.00×10^0	3.80×10^0	3.11×10^0	9.98×10^{-1}	3.43×10^0	9.98×10^{-1}	1.13×10^0	9.98×10^{-1}
	SD	3.20×10^{-17}	2.60×10^0	3.73×10^0	0.00×10^0	3.72×10^0	5.83×10^{-17}	5.03×10^{-1}	1.11×10^{-16}
	Min	1.00×10^0	1.00×10^0	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}	N/A	9.98×10^{-1}
F ₁₅	Mean	1.20×10^{-3}	4.10×10^{-3}	4.36×10^{-3}	4.30×10^{-4}	2.38×10^{-3}	3.07×10^{-4}	3.75×10^{-4}	3.08×10^{-4}
	SD	4.00×10^{-3}	3.20×10^{-3}	8.17×10^{-3}	3.17×10^{-4}	6.03×10^{-3}	6.57×10^{-10}	7.73×10^{-5}	4.22×10^{-8}
	Min	3.10×10^{-3}	1.40×10^{-3}	3.07×10^{-4}	3.07×10^{-4}	3.07×10^{-4}	3.07×10^{-4}	N/A	3.07×10^{-4}
F ₁₆	Mean	-1.00×10^0	-1.00×10^0	-1.02×10^0	-1.03×10^0	-1.03×10^0	-1.03×10^0	-1.03×10^0	-1.03×10^0
	SD	2.30×10^{-16}	4.00×10^{-16}	4.80×10^{-9}	6.78×10^{-6}	3.75×10^{-9}	6.78×10^{-6}	2.39×10^{-6}	0.00×10^0
	Min	-1.00×10^0	-1.00×10^0	-1.03×10^0	-1.03×10^0	-1.03×10^0	-1.03×10^0	N/A	-1.03×10^0
F ₁₇	Mean	4.00×10^{-1}	4.00×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.97×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}
	SD	3.40×10^{-16}	3.40×10^{-16}	3.36×10^{-7}	0.00×10^0	4.86×10^{-7}	0.00×10^0	1.09×10^{-5}	0.00×10^0
	Min	4.00×10^{-1}	4.00×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	N/A	3.98×10^{-1}
F ₁₈	Mean	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0
	SD	3.10×10^{-15}	2.20×10^{-15}	4.88×10^{-6}	1.51×10^{-15}	4.63×10^{-6}	1.24×10^{-14}	9.98×10^{-6}	1.49×10^{-15}
	Min	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0	N/A	3.00×10^0
F ₁₉	Mean	-3.90×10^0	-3.60×10^0	-3.86×10^0	-3.86×10^0	-3.86×10^0	-3.86×10^0	-3.86×10^0	-3.86×10^0
	SD	3.10×10^{-15}	3.00×10^{-1}	1.05×10^{-3}	2.71×10^{-15}	2.71×10^{-3}	2.71×10^{-15}	2.67×10^{-4}	1.51×10^{-15}
	Min	-3.90×10^0	-3.90×10^0	-3.86×10^0	-3.86×10^0	-3.86×10^0	-3.86×10^0	N/A	-3.86×10^0
F ₂₀	Mean	-3.30×10^0	-1.90×10^0	-3.25×10^0	-3.29×10^0	-3.27×10^0	-3.32×10^0	-3.29×10^0	-3.32×10^0
	SD	5.50×10^{-2}	5.40×10^{-1}	7.04×10^{-2}	5.11×10^{-2}	7.37×10^{-2}	3.63×10^{-2}	4.45×10^{-2}	3.83×10^{-7}
	Min	-3.30×10^0	-3.30×10^0	-3.32×10^0	-3.32×10^0	-3.32×10^0	-3.32×10^0	N/A	-3.32×10^0
F ₂₁	Mean	-7.20×10^0	-5.10×10^0	-9.95×10^0	-1.02×10^1	-9.66×10^0	-1.02×10^1	-7.73×10^0	-1.02×10^1
	SD	3.30×10^0	7.40×10^{-3}	1.01×10^0	7.23×10^{-15}	1.51×10^0	3.12×10^{-8}	2.19×10^0	1.55×10^{-3}
	Min	-1.00×10^1	-5.10×10^0	-1.02×10^1	-1.02×10^1	-1.02×10^1	-1.02×10^1	N/A	-1.02×10^1
F ₂₂	Mean	-9.10×10^0	-7.50×10^0	-1.02×10^1	-1.04×10^1	-1.04×10^1	-1.04×10^1	-8.82×10^0	-1.04×10^1
	SD	2.80×10^0	2.70×10^0	1.05×10^0	9.33×10^{-16}	2.66×10^{-4}	3.12×10^{-8}	1.90×10^0	4.40×10^{-2}
	Min	-1.00×10^1	-1.00×10^1	-1.04×10^1	-1.04×10^1	-1.04×10^1	-1.04×10^1	N/A	-1.04×10^1
F ₂₃	Mean	-9.40×10^0	-1.00×10^1	-1.03×10^1	-1.05×10^1	-1.05×10^1	-1.04×10^1	-9.16×10^0	-1.05×10^1
	SD	2.80×10^0	7.80×10^{-1}	1.08×10^0	1.81×10^{-15}	5.41×10^{-1}	7.40×10^{-8}	1.86×10^0	6.64×10^{-10}
	Min	-1.10×10^1	-1.10×10^1	-1.05×10^1	-1.05×10^1	-1.05×10^1	-1.04×10^1	N/A	-1.05×10^1

Table 5. Ranking of the algorithms based on the Friedman test.

Functions	PSO (1998)	GSA (2011)	GWO (2014)	JAYA (2016)	SOGWO (2020)	IGWO (2021)	NCSA (2020)	NGWO (This Work)	p-Value
F ₁	7.000	5.967	4.433	8.000	4.067	3.400	1.833	1.267	4.902×10^{-39}
F ₂	7.060	6.000	4.667	6.877	4.267	2.967	2.000	1.000	8.466×10^{-41}
F ₃	6.830	8.000	2.067	6.100	2.567	4.033	1.400	4.933	8.027×10^{-40}
F ₄	7.000	6.033	3.133	7.967	3.500	4.967	2.400	1.000	4.720×10^{-40}
F ₅	5.400	4.700	5.533	4.367	5.733	2.400	4.767	1.133	1.633×10^{-17}
F ₆	2.000	1.000	6.167	3.000	6.300	4.133	8.000	5.400	1.259×10^{-39}
F ₇	7.900	6.300	3.400	6.800	3.367	4.300	2.400	1.533	2.906×10^{-35}
F ₈	5.133	8.000	6.333	4.267	5.600	2.667	3.000	1.000	1.584×10^{-34}
F ₉	7.767	5.733	2.633	6.933	2.500	5.567	2.433	2.433	2.922×10^{-39}
F ₁₀	7.000	6.000	4.217	8.000	4.417	3.267	1.650	1.450	1.618×10^{-40}
F ₁₁	6.000	8.000	3.133	6.333	3.383	3.617	2.733	2.700	3.735×10^{-33}
F ₁₂	1.767	2.333	5.667	5.567	5.867	3.033	7.567	3.433	3.310×10^{-27}
F ₁₃	2.367	1.300	5.900	4.867	6.267	3.033	7.767	3.533	4.302×10^{-32}
F ₁₄	4.617	6.733	6.067	2.817	5.133	2.817	5.000	2.817	3.877×10^{-20}
F ₁₅	6.500	7.700	4.817	2.333	4.983	2.817	5.213	1.517	1.506×10^{-31}
F ₁₆	3.767	3.767	6.783	3.767	6.617	3.767	3.767	3.767	7.581×10^{-28}
F ₁₇	3.517	3.517	7.567	3.517	7.333	3.517	3.517	3.517	4.257×10^{-40}
F ₁₈	3.500	3.500	7.533	3.500	7.467	3.500	3.500	3.500	7.003×10^{-40}
F ₁₉	3.500	3.500	7.533	3.500	7.467	3.500	3.500	3.500	7.003×10^{-41}
F ₂₀	4.750	2.550	6.700	3.583	6.900	3.433	3.400	2.683	7.641×10^{-23}
F ₂₁	4.133	5.233	5.967	2.500	6.067	2.767	5.250	4.083	3.014×10^{-15}
F ₂₂	3.483	2.983	6.867	2.983	6.967	3.050	4.750	4.917	3.309×10^{-25}
F ₂₃	4.650	1.500	6.633	1.500	6.867	4.033	5.500	5.317	2.890×10^{-31}

3.2. Classical Benchmark Functions

3.2.1. Exploitation Analysis

The unimodal functions (F_1 – F_7) are well recognized for evaluating algorithm exploitation ability. The performance indexes of unimodal functions are reported in Table 3. The proposed NGWO shows the superior result for the considered unimodal functions and outperforms all other algorithms for five unimodal functions (F_1 , F_2 , F_4 , F_5 , F_7). Nonetheless, the NCSA algorithm performs better for one unimodal function (F_3), and GSA achieved the best indexes for one unimodal function (F_6).

3.2.2. Exploration Analysis and Local Optima Avoidance

The multi-modal functions are renowned for having multiple local optima, where the number of local optima increases exponentially with dimensions. Therefore, the multi-modal functions (F_8 – F_{23}) are suitable for testing the exploration and local optima avoidance ability of an algorithm. The NGWO algorithm shows very promising results for the multi-modal functions. As seen from Table 4, the NGWO algorithm outperforms all other algorithms and accomplished all the best indexes for nine functions (F_8 , F_9 , F_{11} , F_{16} – F_{20} , F_{23}). Moreover, the NGWO algorithm found the global optima of ten multi-modal functions with pinpoint precision (F_9 , F_{11} , F_{16} – F_{23}). The NGWO algorithm achieved the best SD indexes for a total of 14 functions, indicating the algorithm's high local optima avoidance ability. The lowest SD values, therefore, mean that there is less variation in the algorithm's optimum values, implying that the algorithm is stable.

3.2.3. Convergence Analysis

This paragraph illustrates the convergence ability of the NGWO and other compared algorithms. Figure 2 shows the convergence characteristics of the unimodal benchmarks, while Figure 3 depicts the convergence behavior of the six multi-modal benchmarks. In Figures 2 and 3, the horizontal axis signifies the iteration number, and vertical axis shows the median function values from 30 runs. In all Figures, the NGWO maintains a consistent downward slope through the iteration periods, which indicates that the wolves are sharing information to improve the cost function values (fitness) through the iteration period. For most of the convergence curves of Figures 2 and 3, the NGWO shows superior convergence ability that also indicates a proper balance between exploitation and exploration. Furthermore, the NGWO algorithm only took 100 iterations to achieve the exact global optima zero for multi-modal functions F_9 and F_{11} , demonstrating NGWO's rapid convergence speed.

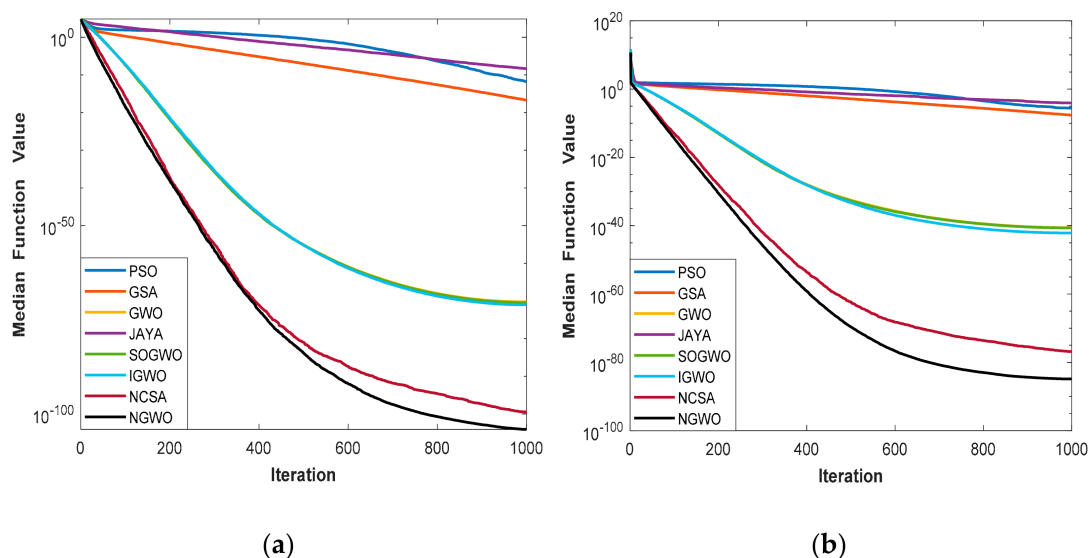


Figure 2. Cont.

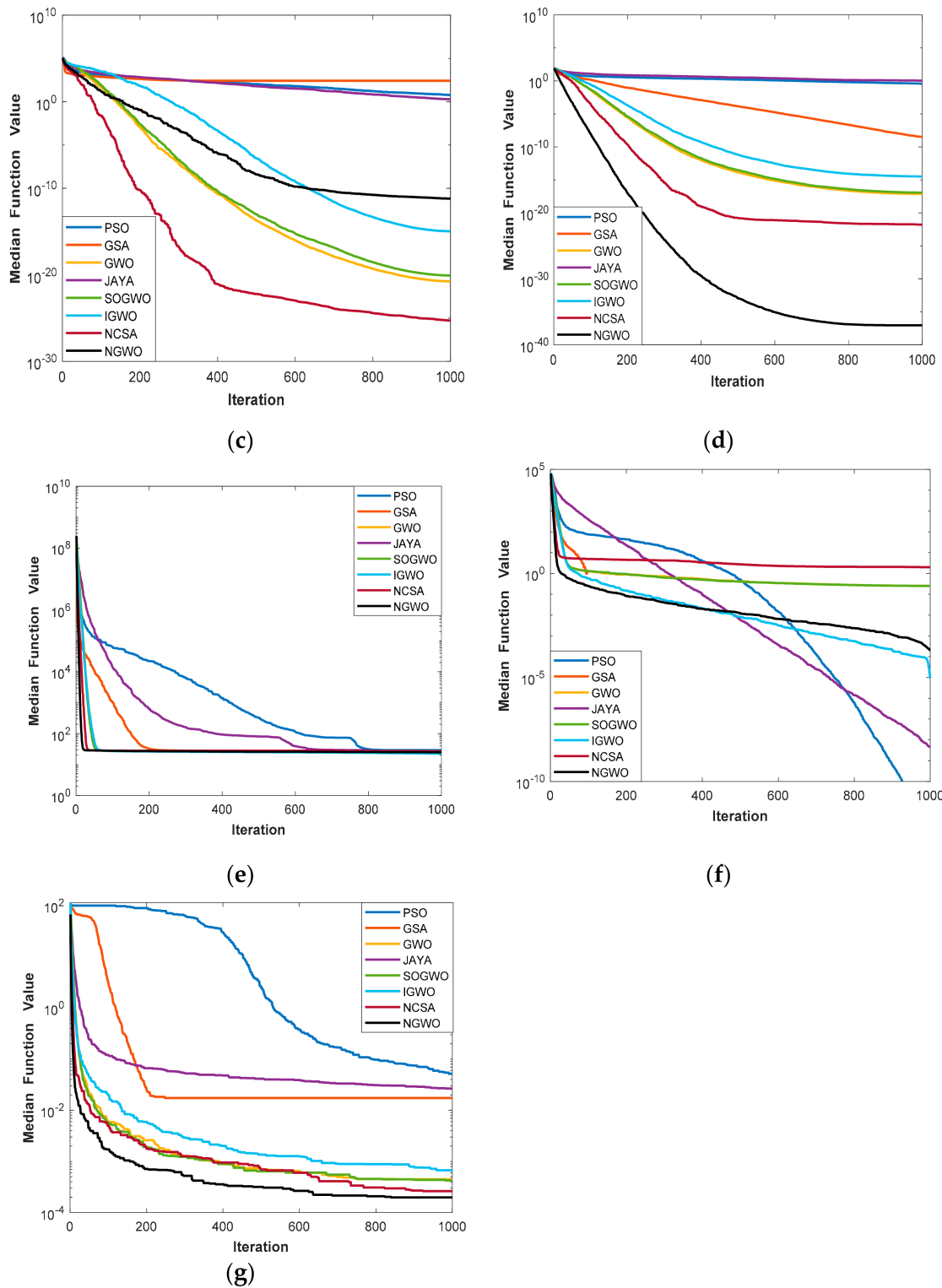


Figure 2. Convergence characteristics of the algorithms for unimodal benchmark functions (F_1 – F_7). Here, (a) represents benchmark function F_1 , (b) represents benchmark function F_2 , (c) represents benchmark function F_3 , (d) represents benchmark function F_4 , (e) represents benchmark function F_5 , (f) represents benchmark function F_6 , (g) represents benchmark function F_7 .

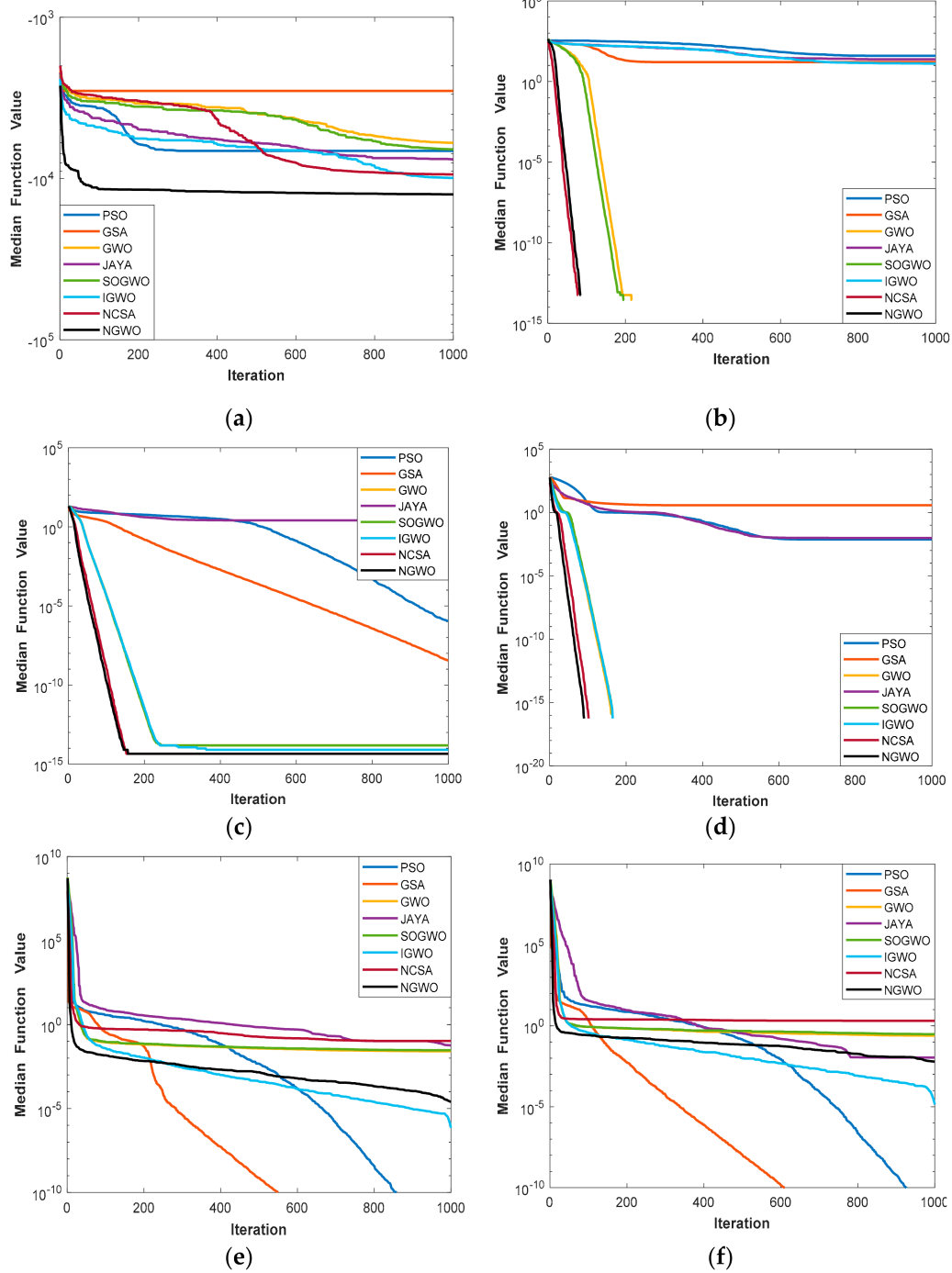


Figure 3. Convergence characteristics of the algorithms for multi-modal benchmark functions (F_8 – F_{13}). Here, (a) represents benchmark function F_8 , (b) represents benchmark function F_9 , (c) represents benchmark function F_{10} , (d) represents benchmark function F_{11} , (e) represents benchmark function F_{12} , (f) represents benchmark function F_{13} .

The parameter space, quest history, trajectory of the first wolf, convergence curves, and wolves average fitness are plotted in Figures 4 and 5 to illustrate the convergence behavior in detail. This section has considered the number of wolves and iteration as 50 and 100, respectively. It has been known from the literature [52] that abrupt position changes during the initial iteration steps are more preferable for improving the exploration ability of an algorithm. These changes need to be decreased to strengthen the exploitation ability in the later part of the iteration. The trajectory and search history of the 1st wolf are demonstrated in Figures 4 and 5. It is explicit from the first wolf’s search history (second

column of Figures 4 and 5) that the wolves extensively searched through the search space and exploited the best area. The trajectory of 1st wolf (3rd column in Figures 4 and 5) also indicates the same phenomenon, where it emphasized more on exploration in the initial steps, and it decreased through the iteration period. By contrast, the exploitation ability is strengthened during the last periods of iteration. All of these indicate that the NGWO has good convergence ability, and it achieves a proper balance between local exploitation and global exploration.

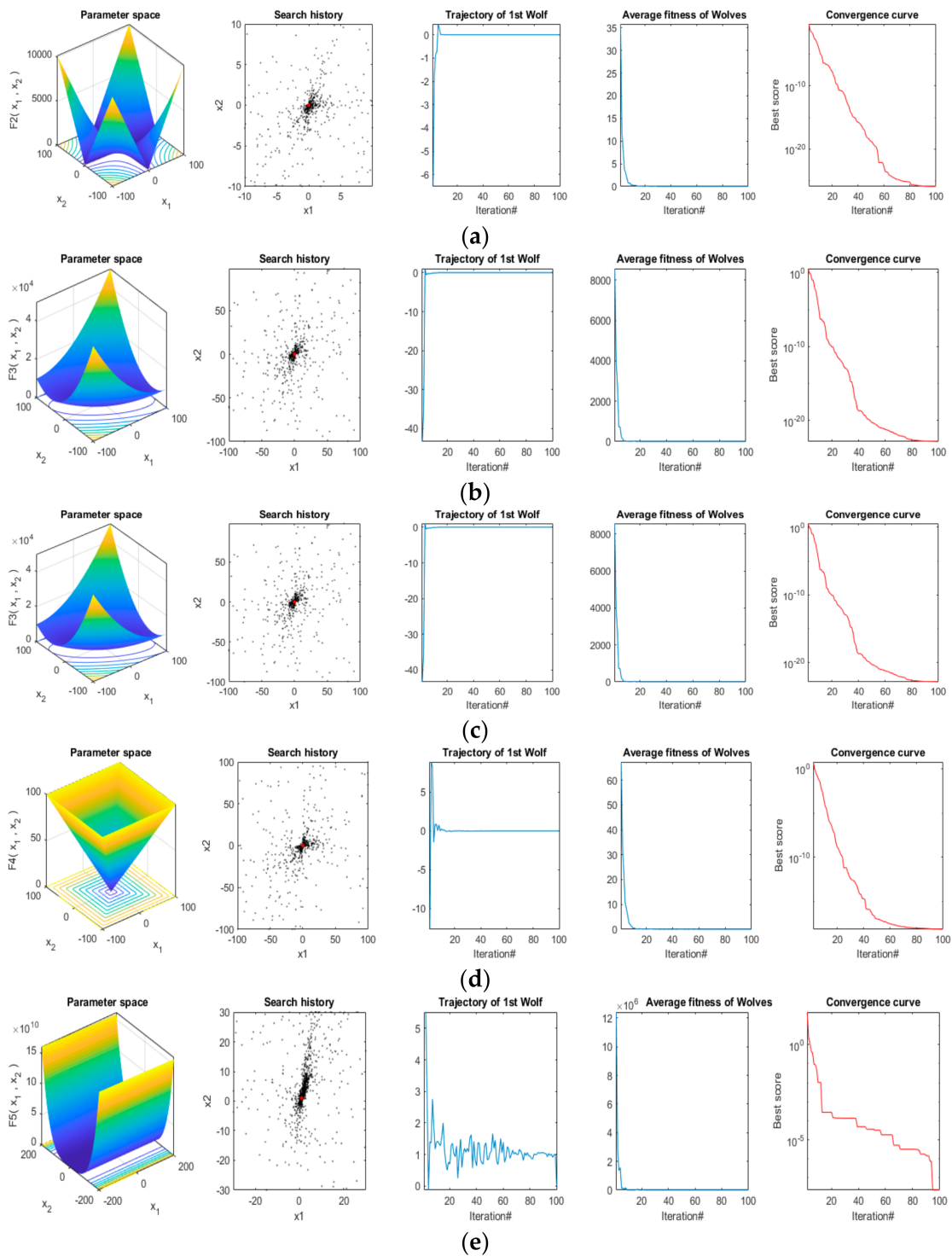


Figure 4. Cont.

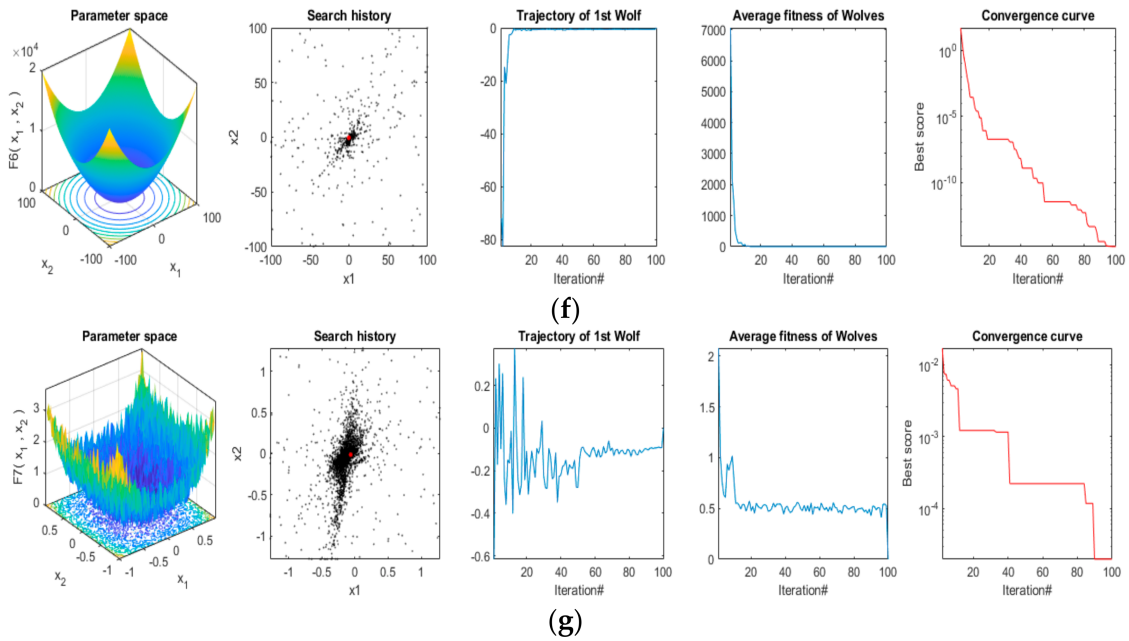


Figure 4. Parameter space, search history, trajectory of 1st wolf, the average fitness of all wolves, and convergence curve of NGWO for unimodal functions (F_1 – F_7). Here, (a) represents benchmark function F_1 , (b) represents benchmark function F_2 , (c) represents benchmark function F_3 , (d) represents benchmark function F_4 , (e) represents benchmark function F_5 , (f) represents benchmark function F_6 , (g) represents benchmark function F_7 .

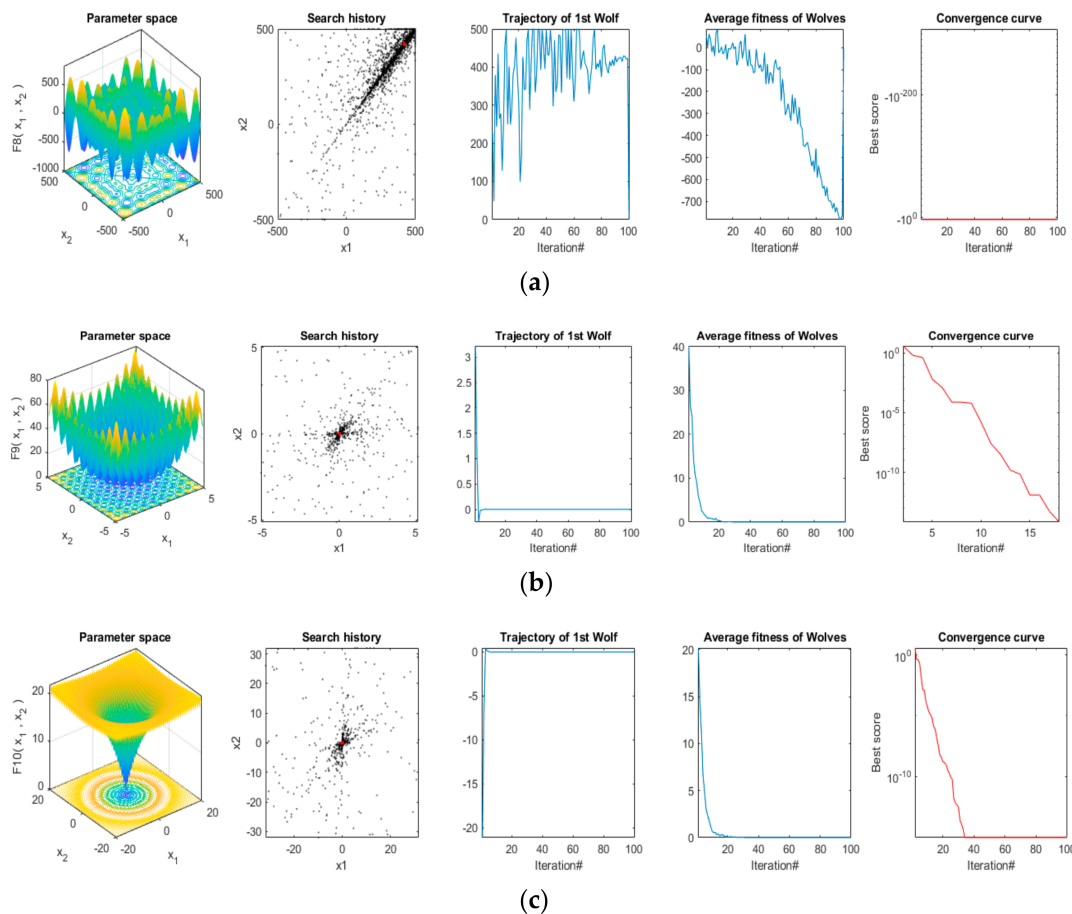


Figure 5. Cont.

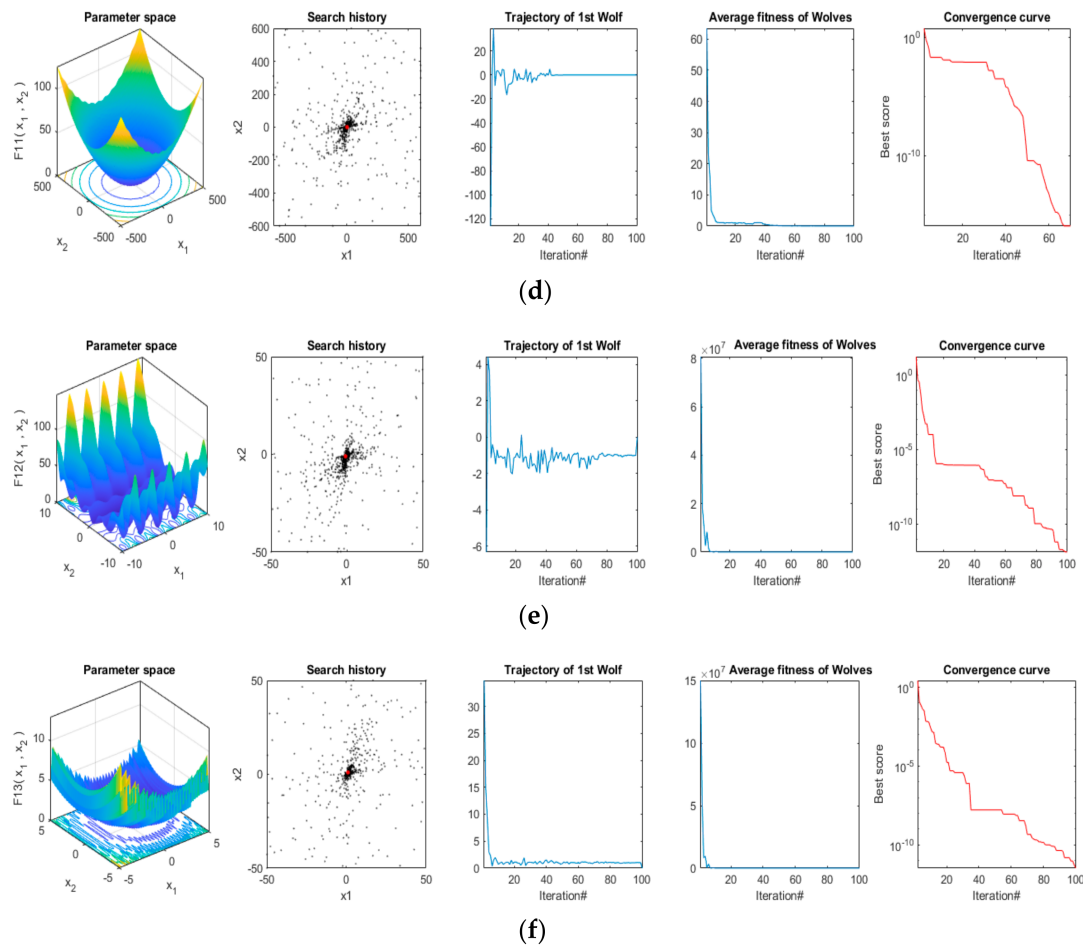


Figure 5. Parameter space, search history, trajectory of 1st wolf, the average fitness of wolves, and convergence curve of NGWO for multi-modal functions (F_8 – F_{13}). Here, (a) represents benchmark function F_8 , (b) represents benchmark function F_9 , (c) represents benchmark function F_{10} , (d) represents benchmark function F_{11} , (e) represents benchmark function F_{12} , (f) represents benchmark function F_{13} .

Figures 6 and 7 show the box plots for each algorithm for multi-modal functions. The first six multi-modal functions are 30-dimensional problems (F_7 – F_{13}), while the next ten are fixed dimensional multi-modal problems, and the dimension numbers are reported in Table 1. In MATLAB, we run each algorithm 30 times for each test function, saving the best values from each run and plotting them as box plots. These box plots illustrate the distribution of benchmark function results in a better way and indicates the dispersion and symmetry of data sets. The box plots represent the first, second (median), third and lower quartiles, upper bounds and outliers, respectively. The bigger size of the rectangular box indicates that the data sets are scattered more, and the results obtained by the algorithm are not stable. It also indicates the comparatively worst performance. The red crossbar indicates the median of data sets, and the plus sign (+) in red indicates outliers. Among the plots of Figure 6, the box plot location of the NGWO is lower than the other seven algorithms for four functions (F_8 – F_{11} , Figure 6a–d), and the box size is also lowest for these functions. In Figure 6b,d the NGWO algorithm reached the exact zero for all data sets, that is why there is no box for NGWO in these two figures. In the case of Figure 7a–j, the NGWO shows the minimum distribution of the data sets and its box plots lies below that of other algorithms. These things indicate the superiority of the proposed NGWO algorithm and less variation in results, and stability of the algorithm.

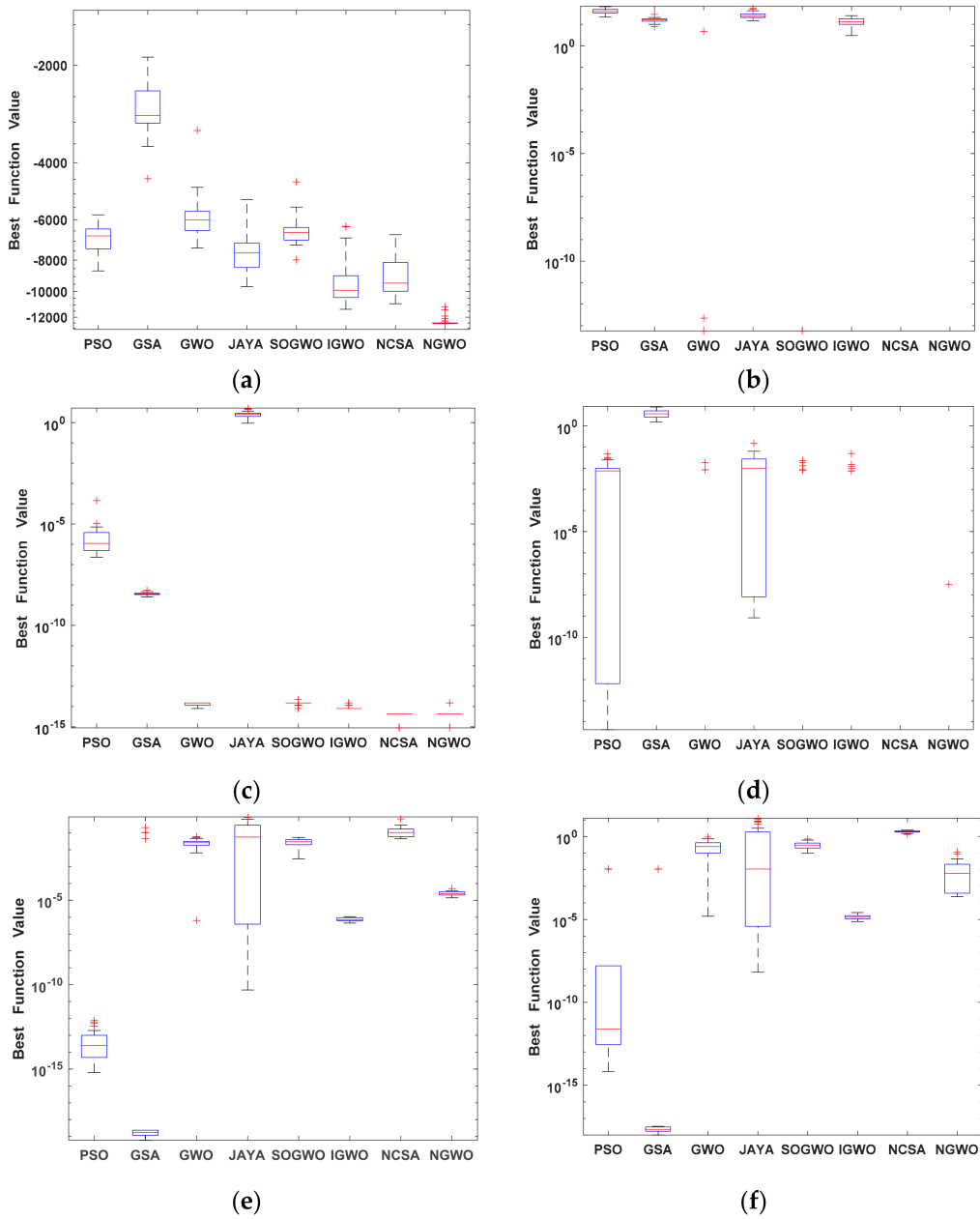


Figure 6. Box plot of the multi-modal benchmark functions (F_8 – F_{13}). Here, (a) represents F_8 , (b) represents F_9 , (c) represents F_{10} , (d) represents F_{11} , (e) represents F_{12} , (f) represents F_{13} .

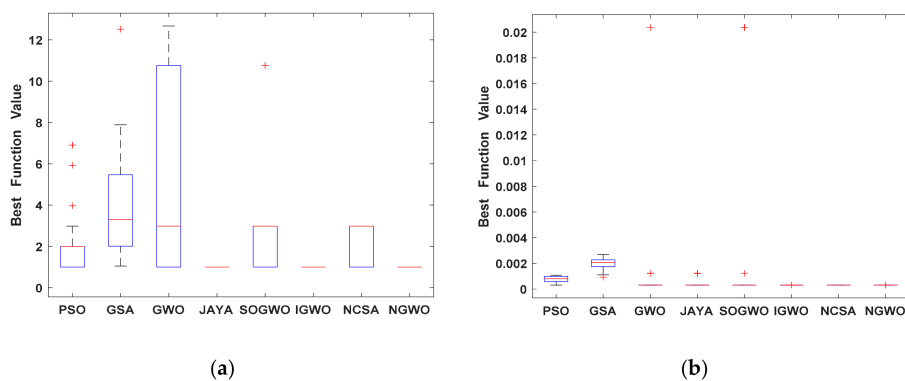


Figure 7. Cont.

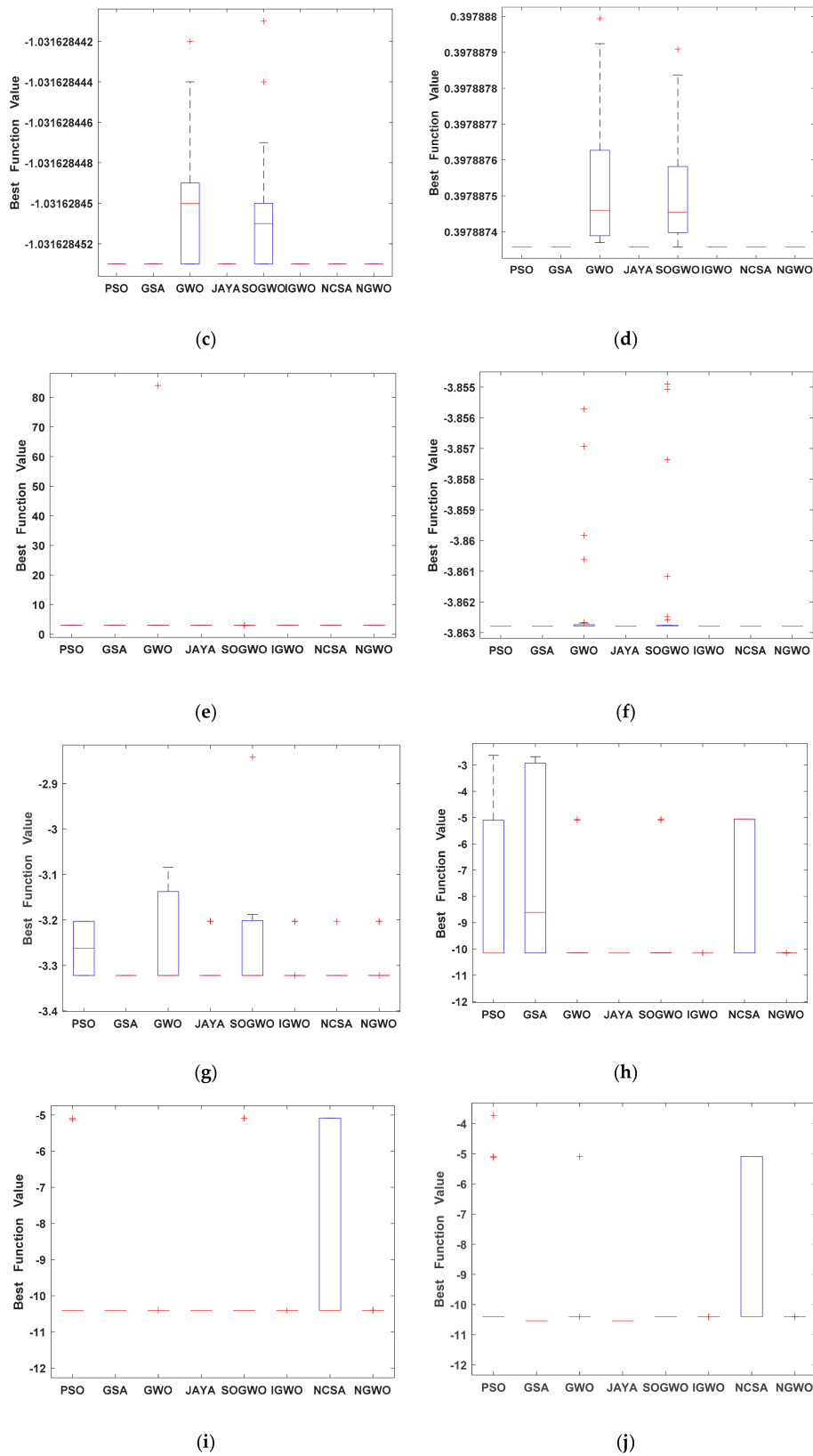


Figure 7. Box plot of the fixed dimensional multi-modal benchmark functions (F_{14} – F_{23}). Here, (a) represents benchmark function F_{14} , (b) represents benchmark function F_{15} , (c) represents benchmark function F_{16} , (d) represents benchmark function F_{17} , (e) represents benchmark function F_{18} , (f) represents benchmark function F_{19} , (g) represents benchmark function F_{20} , (h) represents benchmark function F_{21} , (i) represents benchmark function F_{22} , (j) represents benchmark function F_{23} .

3.2.4. Ranking of the Algorithms

To analyze the NGWO algorithm's performance more thoroughly, we conducted non-parametric Friedman tests to rank each algorithm. This multiple comparison test works based on the recurrent measurement of the Anova, and it can determine the differences among the compared algorithms. The best values of each algorithm for 30 runs were stored in a matrix, and the Friedman test was applied to this matrix to obtain the mean ranks of each algorithm for 30 runs. The Friedman test also provides the corresponding p -values for each test function. It is explicit from the Friedman test results in Table 5 that the proposed NGWO attained the best rank for 16 functions among a total of 23 benchmark functions. Additionally, the JAYA, GSA, and PSO obtained the best ranking for 3, 2, and 1 functions, respectively. Very low p -values indicate that the Friedman test results are statistically significant.

3.3. Engineering Application of NGWO

The proposed algorithm has been tested for three real-life engineering case studies and collected from the literature [18,43]. All three problems have inequality constraints, and we use the hard penalty approach to handle the constraints. We have considered 100 wolves over 100 iterations to solve these three problems.

3.3.1. Welded Beam Design

The objective of this case study is to minimize a welded beam's fabrication cost. It consists of four decision variables such as bar thickness (b), bar height (t), attached part length (l), and weld thickness (h), and seven inequality constraints (shear and bending stress, side constraints, end deflection, buckling load). The problem is mathematically formulated as [43]:

$$f(y) = 1.10471y_1^2y_2 + 0.04811y_3y_4(14 + y_2) \quad (34)$$

$$\tau(y) = \sqrt{\tau_1^2 + \tau_1 \tau_2 \frac{y_2}{2R} + \tau_2^2} \quad (35)$$

$$\tau_1 = \frac{P}{\sqrt{2}y_1y_2} \quad (36)$$

$$\tau_2 = \frac{MR}{J} \quad (37)$$

$$M = P\left(L + \frac{y_2}{2}\right) \quad (38)$$

$$R = \sqrt{0.25 * (y_2^2 + (y_1 + y_3)^2)} \quad (39)$$

$$J = 2 / \left\{ \sqrt{2}y_1y_2 \left[\frac{y_2^2}{12} + \frac{(y_1 + y_3)^2}{4} \right] \right\} \quad (40)$$

$$\sigma(y) = \frac{6PL}{x_4y_3^2} \quad (41)$$

$$\delta(y) = \frac{4PL^3}{Ey_4y_3^3} \quad (42)$$

$$P_c(y) = \frac{4.013E y_3y_4^3}{6L^2} \left(1 - 0.25y_3 \sqrt{\left(\frac{E}{G}\right)/L}\right) \quad (43)$$

Here, $P = 6000\text{lb}$, $L = 14\text{ inch}$, $\delta_{\max} = 0.25\text{ inch}$, $E = 30 \times 10^6\text{ psi}$, $G = 12 \times 10^6\text{ psi}$, $\tau_{\max} = 13600\text{ psi}$, $\sigma_{\max} = 30000\text{ psi}$. The inequality constraints are:

$$c_1(y) = \tau(y) - \tau_{\max} \leq 0 \quad (44)$$

$$c_2(y) = \sigma(y) - \sigma_{\max} \leq 0 \quad (45)$$

$$c_3(y) = y_1 - y_4 \leq 0 \quad (46)$$

$$c_4(y) = 1.10471y_1^2 + 0.04811y_3y_4(14 + y_2) - 5 \leq 0 \quad (47)$$

$$c_5(y) = \delta(y) - \delta_{\max} \leq 0 \quad (48)$$

$$c_6(y) = p - p_c(y) \leq 0 \quad (49)$$

$$c_7(y) = 0.125 - y_1 \leq 0 \quad (50)$$

The decision variable bounds are:

$$0.1 \leq y_1 \leq 2; 0.1 \leq y_2 \leq 10; 0.1 \leq y_3 \leq 10; 0.1 \leq y_4 \leq 2$$

3.3.2. Tension–Compression Coil Spring Design

The weight of the tension–compression coil is reduced in this case study. It consists of three decision variables ((number of coils (n), wire diameter (d), mean coil diameter (D)), and 4 inequality constraints. The following is a numerical representation of the problem:

$$\text{Minimize } f(y) = (y_3 + 2) y_1 y_2^2$$

The inequality constraints are:

$$c_1(y) = 1 - \frac{y_1^3 y_3}{71785 y_2^4} \leq 0 \quad (51)$$

$$c_2(y) = \frac{4y_1^2 - y_1 y_2}{12566(y_1 y_2^3 - y_2^4)} + \frac{1}{5108 y_2^2} - 1 \leq 0 \quad (52)$$

$$c_3(y) = 1 - \frac{140.45 y_2}{y_1^2 y_3} \leq 0 \quad (53)$$

$$c_4(y) = \frac{y_1 + y_2}{1.5} - 1 \leq 0 \quad (54)$$

Here, $0.25 \leq y_1 \leq 1.30$; $0.05 \leq y_2 \leq 2$; $2 \leq y_3 \leq 15$;

3.3.3. Three Bars Truss Design

Ray and Saini [53] first proposed this case study to minimize the bar's weight. It has 2 variables and 3 constraints, and it can be mathematically presented as:

$$\text{Minimize } f(y) = (2\sqrt{2} y_1 + y_2) \times l$$

Subject to

$$\frac{\sqrt{2} y_1 + y_2}{\sqrt{2} y_1^2 + 2 y_1 y_2} (P - \sigma) \leq 0 \quad (55)$$

$$\frac{y_2}{\sqrt{2} y_1^2 + 2 y_1 y_2} (P - \sigma) \leq 0 \quad (56)$$

$$\frac{1}{2 y_1 y_2 + \sqrt{2} y_1^2} (P - \sigma) \leq 0 \quad (57)$$

The variables ranges are: $0 \leq y_1 \leq 1$; $0 \leq y_2 \leq 1$ and $l = 100\text{ cm}$, $P = 2\text{ KN/cm}^2$, and $\sigma = 2\text{ KN/cm}^2$

3.4. Results of the Engineering Application

3.4.1. Optimization Result of Welded Beam Design

It is explicit from the reported results shown in Table 6 that the NGWO outperforms all the previous algorithms for this case study. The proposed algorithm achieved optimal values for four indexes among the five compared indexes (best, mean, worst, standard deviation, and NFEs). In contrast, the HEAA algorithm achieved the best result for the standard deviation index.

Table 6. Comparative study of Welded Beam design problem.

Algorithms	y ₁	y ₂	y ₃	y ₄	f (Best)	f (Mean)	f (Worst)	SD	NFEs
NGWO	0.34094	3.5810	9.0321	0.2063	2.036	2.046	2.0632	1.02 × 10 ²	20,100
GA [54]	0.2489	6.173	8.1789	0.2533	2.4331	N/A	N/A	N/A	N/A
SC [55]	0.2444	6.238	8.2886	0.2446	2.3854	3.2551	6.3997	9.60 × 10 ⁻¹	33,095
FSA [56]	0.2443	6.2158	8.2939	0.2443	2.3811	2.4042	2.489	N/A	56,243
AATM [57]	0.2441	6.2209	8.2982	0.2444	2.3823	2.387	2.3916	2.20 × 10 ⁻³	30,000
HEAA [58]	0.2444	6.2175	8.2915	0.2444	2.381	2.381	2.381	1.30 × 10⁻⁵	30,000
EEGWO [41]	0.2444	6.217	8.2928	0.2444	2.3813	2.3817	2.3824	4.18 × 10 ⁻⁴	50,000

3.4.2. Result of Tension–Compression Coil Spring Design

For tension–compression coil spring design, we compared our findings with 10 other previous studies from the literature in Table 7. The NGWO algorithm achieved the optimal values for four indexes, while the FSA algorithm attained the optimal value for SD index.

Table 7. Comparative study of tension–compression coil spring design.

Algorithms	y ₁	y ₂	y ₃	f (Best)	f (Mean)	f (Worst)	SD	NFEs
NGWO	10.32826	0.35486	0.05026	0.011053	0.011755	0.012454	5.21 × 10 ⁻⁴	20,100
GA [59]	10.89052	0.36396	0.05198	0.01268	0.01274	0.01297	5.90 × 10 ⁻⁵	80,000
FSA [56]	11.21390	0.35800	0.05174	0.01266	0.01266	0.01266	2.20 × 10⁻⁸	49,531
CPSO [60]	11.24454	0.35764	0.05172	0.01267	0.01273	0.01292	5.20 × 10 ⁻⁵	200,000
HPSO [60]	11.26503	0.35712	0.05170	0.01266	0.01270	0.01271	1.58 × 10 ⁻⁵	81,000
GSA [13]	14.22867	0.05	0.31731	0.01287	0.01343	0.01421	1.34 × 10 ⁻²	30,000
GWO [18]	12.04249	0.34454	0.05117	0.01267	0.01269	0.01272	2.10 × 10 ⁻⁵	30,000
AFA [61]	11.31956	0.35619	0.05166	0.01266	0.01266	0.01280	1.27 × 10 ⁻²	50,000
TEO [62]	11.16839	0.35879	0.05177	0.01266	0.01268	0.01271	4.41 × 10 ⁻²	300,000
MGWO [63]	11.80809	0.34819	0.05133	0.01266	0.01267	0.01270	1.10 × 10 ⁻⁵	30,000
EEGWO [41]	11.3113	0.35634	0.05167	0.01266	0.01268	0.01272	2.22 × 10 ⁻⁵	50,000

3.4.3. Three Bars Truss Design Result

Table 8 summarizes the results of three bars truss design problem. The WCA and DEDS algorithms achieved three and two best indexes, respectively. The NGWO algorithm’s results are very similar to the previous best results obtained by various optimization algorithms.

Table 8. Results of three bars truss design problem.

Algorithms	y ₁	y ₂	f (Best)	f (Mean)	f (Worst)	SD	NFEs
NGWO	0.789186	0.406806	263.8959	263.8964	263.8971	5.58×10^{-4}	20,100
PSO [6]	0.781224	0.432548	264.2183	265.9553	267.459	1.38×10^0	30,000
SC [55]	0.788621	0.408401	263.8958	263.9033	263.9697	1.30×10^{-2}	17,610
DEDS [64]	0.788651	0.408316	263.8958	263.8958	263.8958	9.70×10^{-7}	15,000
GSA [13]	0.777662	0.448853	264.8299	271.0348	279.7925	4.12×10^0	30,000
HEAA [58]	0.78868	0.408234	263.8958	263.8959	263.8961	4.90×10^{-5}	15,000
AATM [57]	0.788682	0.408229	263.8958	263.8966	263.9004	1.10×10^{-3}	17,000
WCA [65]	0.788651	0.408316	263.8958	263.8959	263.8962	8.71×10^{-5}	5250
MBA [66]	0.788565	0.40856	263.8959	263.898	263.916	3.93×10^{-3}	13,280
GWO [18]	0.788409	0.409003	263.8959	263.8966	263.898	4.37×10^{-4}	30,000
MVO [67]	0.788993	0.407351	263.8959	263.8961	263.8971	2.49×10^{-4}	30,000
SCA [68]	0.789068	0.407162	263.8984	263.9356	263.9951	2.88×10^{-2}	30,000
MGWO [63]	0.788561	0.408572	263.8959	263.8963	263.8976	4.29×10^{-4}	30,000
EEGWO [41]	0.78841	0.40899	263.896	263.8963	263.8966	2.19×10^{-4}	50,000

4. Conclusions

The GWO algorithm was influenced by the leadership structure and group hunting system of wolves. The GWO algorithm undergoes the lack of population diversity and is vulnerable to premature convergence in multi-modal problems. Hence, a new variant of GWO called niching GWO (NGWO) with a modified position update equation is proposed in this research to improve the performance of the GWO algorithm for solving multi-modal problems. The niching techniques are well known for handling multiple optimal solutions in the case of multi-modal optimization problems. A fitness Euclidean distance ratio-based niching GWO called FER-GWO is used to strengthen the searchability of the algorithm. By using the FER technique, the wolves are guided towards the nearby wolves' best position, having better cost function values.

The proposed NGWO algorithm has been thoroughly tested on 23 traditional benchmark functions and three real-world engineering cases. It achieves very promising results and outperforms most other compared algorithms for the considered benchmarks and engineering case studies. This superiority is accomplished by incorporating the niching technique to the basic GWO algorithm and modifying the position update equation, whereas the local search increases the fine searchability. The reported results indicate that the algorithm achieved well-balanced global exploration and local exploitation, which further increases the algorithm's accuracy. This research focuses on single-objective optimization problems only; however, future work will extend the NGWO algorithm to multi-objective optimization problems.

Author Contributions: Conceptualization, R.A. and S.M.; methodology, R.A., S.M., M.S. and A.N.; software, R.A.; validation, A.N., R.A., S.M. and J.I.; formal analysis, R.A., J.I.; investigation, R.A., J.I.; resources, R.A., A.N., M.S.; data curation, R.A.; writing—original draft preparation, R.A. S.M.; writing—review and editing, A.N., M.S., J.I.; visualization, R.A., J.I.; supervision, S.M., A.N.; project administration, A.N.; S.M., M.S.; funding acquisition, A.N.; M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Zayed University, Abu Dhabi, UAE, and in part by the AnalytiCray Solutions, Kuala Lumpur, Malaysia, and in part by the Taif University Researchers Supporting Project number (TURSP-2020/79), Taif University, Taif, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Xu, X.; Tang, Y.; Li, J.; Hua, C.; Guan, X. Dynamic multi-swarm particle swarm optimizer with cooperative learning strategy. *Appl. Soft Comput.* **2015**, *29*, 169–183. [[CrossRef](#)]
2. Qu, B.; Liang, J.; Suganthan, P. Niching particle swarm optimization with local search for multi-modal optimization. *Inf. Sci.* **2012**, *197*, 131–143. [[CrossRef](#)]
3. Al Amin, M.; Abdul-Rani, A.M.; Ahmed, R.; Rao, T.V.V.L.N. Multiple-objective optimization of hydroxyapatite-added EDM technique for processing of 316L-steel. *Mater. Manuf. Process.* **2021**, *36*, 1–12. [[CrossRef](#)]
4. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; The MIT Press: Cambridge, MA, USA, 1975.
5. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [[CrossRef](#)]
6. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS'95, Nagoya, Japan, 4–6 October 1995; pp. 39–43. [[CrossRef](#)]
7. Storn, R.; Price, K. Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces. *J. Glob. Optim.* **1995**, *23*, 1–15.
8. Dorigo, M.; Gambardella, L. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1997**, *1*, 53–66. [[CrossRef](#)]
9. Lu, X.; Zhou, Y. A novel global convergence algorithm: Bee collecting pollen algorithm. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence. ICIC 2008. Lecture Notes in Computer Science*; Huang, D.S., Wunsch, D.C., Levine, D.S., Jo, K.H., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5227. [[CrossRef](#)]
10. Formato, R.A. Central force optimization: A new metaheuristic with applications in applied electromagnetics. *Prog. Electromagn. Res.* **2007**, *77*, 425–491. [[CrossRef](#)]
11. Hatamlou, A. Black hole: A new heuristic optimization approach for data clustering. *Inf. Sci.* **2013**, *222*, 175–184. [[CrossRef](#)]
12. Tabari, A.; Ahmad, A. A new optimization method: Electro-Search algorithm. *Comput. Chem. Eng.* **2017**, *103*, 1–11. [[CrossRef](#)]
13. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [[CrossRef](#)]
14. Rao, R.; Savsani, V.; Vakharia, D. Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Comput. Aided Des.* **2011**, *43*, 303–315. [[CrossRef](#)]
15. Alatas, B. ACROA: Artificial Chemical Reaction Optimization Algorithm for global optimization. *Expert Syst. Appl.* **2011**, *38*, 13170–13180. [[CrossRef](#)]
16. Yang, X.S. Firefly algorithm, stochastic test functions and design optimization. *Int. J. Bio Inspired Comput.* **2010**, *2*, 78. [[CrossRef](#)]
17. Erol, O.K.; Eksin, I. A new optimization method: Big Bang-Big Crunch. *Adv. Eng. Softw.* **2006**, *37*, 106–111. [[CrossRef](#)]
18. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
19. 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](#)]
20. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Futur. Gener. Comput. Syst.* **2019**, *97*, 849–872. [[CrossRef](#)]
21. Rao, R.V. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int. J. Ind. Eng. Comput.* **2016**, *7*, 19–34. [[CrossRef](#)]
22. Tu, Q.; Chen, X.; Liu, X. Hierarchy strengthened Grey Wolf Optimizer for numerical optimization and feature selection. *IEEE Access* **2019**, *7*, 78012–78028. [[CrossRef](#)]
23. Al-Tashi, Q.; Rais, H.M.; Abdulkadir, S.J.; Mirjalili, S. Feature selection based on Grey Wolf Optimizer for oil & gas reservoir classification. In Proceedings of the 2020 International Conference on Computational Intelligence (ICCI), Seri Iskandar, Perak, Malaysia, 8–9 October 2020; pp. 211–216. [[CrossRef](#)]
24. Emary, E.; Zawbaa, H.M.; Grosan, C.; Hassenian, A.E. Feature subset selection approach by gray-wolf optimization. In *Advances in Intelligent Systems and Computing*; Springer: Cham, Switzerland, 2015; Volume 334. [[CrossRef](#)]
25. Song, J.; Wang, J.; Lu, H. A novel combined model based on advanced optimization algorithm for short-term wind speed forecasting. *Appl. Energy* **2018**, *215*, 643–658. [[CrossRef](#)]
26. El-Fergany, A.A.; Hasani, H.M. Single and multi-objective optimal power flow using Grey Wolf Optimizer and differential evolution algorithms. *Electr. Power Compon. Syst.* **2015**, *43*, 1548–1559. [[CrossRef](#)]
27. Sulaiman, M.H.; Mustaffa, Z.; Mohamed, M.R.; Aliman, O. Using the gray wolf optimizer for solving optimal reactive power dispatch problem. *Appl. Soft Comput. J.* **2015**, *32*, 286–292. [[CrossRef](#)]
28. Katarya, R.; Verma, O.P. Recommender system with Grey Wolf Optimizer and FCM. *Neural Comput. Appl.* **2018**, *30*, 1679–1687. [[CrossRef](#)]
29. Kamboj, V.K. A novel hybrid PSO–GWO approach for unit commitment problem. *Neural Comput. Appl.* **2016**, *27*, 1643–1655. [[CrossRef](#)]
30. Panwar, L.K.; Reedy, S.; Verma, A.; Panigrahi, B.K.; Kumar, R. Binary Grey Wolf Optimizer for large scale unit commitment problem. *Swarm Evol. Comput.* **2018**, *38*, 251–266. [[CrossRef](#)]
31. Biswas, K.; Vasant, P.M.; Vintaned, J.A.G.; Watada, J. Cellular automata-based multi-objective hybrid Grey Wolf Optimization and particle swarm optimization algorithm for wellbore trajectory optimization. *J. Nat. Gas Sci. Eng.* **2021**, *85*, 103695. [[CrossRef](#)]

32. Korayem, L.; Khorsid, M.; Kassem, S.S. Using Grey Wolf algorithm to solve the capacitated vehicle routing problem. *Conf. Ser. Mater. Sci. Eng.* **2015**, *2015*, 83. [[CrossRef](#)]
33. Lu, C.; Xiao, S.; Li, X.; Gao, L. An effective multi-objective discrete grey Wolf optimizer for a real-world scheduling problem in welding production. *Adv. Eng. Softw.* **2016**, *99*, 161–176. [[CrossRef](#)]
34. Komaki, G.; Kayvanfar, V. Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time. *J. Comput. Sci.* **2015**, *8*, 109–120. [[CrossRef](#)]
35. Asadzadeh, L. A local search genetic algorithm for the job shop scheduling problem with intelligent agents. *Comput. Ind. Eng.* **2015**, *85*, 376–383. [[CrossRef](#)]
36. Jayabarathi, T.; Raghunathan, T.; Adarsh, B.; Suganthan, P.N. Economic dispatch using hybrid grey wolf optimizer. *Energy* **2016**, *111*, 630–641. [[CrossRef](#)]
37. Pradhan, M.; Roy, P.K.; Pal, T. Grey Wolf optimization applied to economic load dispatch problems. *Int. J. Electr. Power Energy Syst.* **2016**, *83*, 325–334. [[CrossRef](#)]
38. Guha, D.; Roy, P.K.; Banerjee, S. Load frequency control of large scale power system using quasi-oppositional Grey Wolf Optimization algorithm. *Eng. Sci. Technol. Int. J.* **2016**, *19*, 1693–1713. [[CrossRef](#)]
39. Singh, N.; Singh, S. A novel hybrid GWO-SCA approach for optimization problems. *Eng. Sci. Technol. Int. J.* **2017**, *20*, 1586–1601. [[CrossRef](#)]
40. Jiang, T.; Zhang, C. Application of Grey Wolf Optimization for solving combinatorial problems: Job Shop and Flexible Job Shop scheduling cases. *IEEE Access* **2018**, *6*, 26231–26240. [[CrossRef](#)]
41. Long, W.; Jiao, J.; Liang, X.; Tang, M. An exploration-enhanced Grey Wolf Optimizer to solve high-dimensional numerical optimization. *Eng. Appl. Artif. Intell.* **2018**, *68*, 63–80. [[CrossRef](#)]
42. Dhargupta, S.; Ghosh, M.; Mirjalili, S.; Sarkar, R. Selective opposition based Grey Wolf Optimization. *Expert Syst. Appl.* **2020**, *151*, 113389. [[CrossRef](#)]
43. Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S. An improved Grey Wolf Optimizer for solving engineering problems. *Expert Syst. Appl.* **2021**, *166*, 113917. [[CrossRef](#)]
44. Yin, X.; Gernay, N. A Fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. In *Artificial Neural Nets and Genetic Algorithms*; Springer: Cham, Switzerland, 1993.
45. Mahfound, S.W. Crowding and preselection revisited. In *Parallel Problem Solving from Nature*; Elsevier: Amsterdam, The Netherlands, 1992.
46. Petrowski, A. Clearing procedure as a niching method for genetic algorithms. In Proceedings of the IEEE International Conference on Evolutionary Computation, Nayoya, Japan, 20–22 May 1996. [[CrossRef](#)]
47. Li, X. Niching without niching parameters: Particle swarm optimization using a ring topology. *IEEE Trans. Evol. Comput.* **2010**, *14*, 150–169. [[CrossRef](#)]
48. Li, X. A multimodal particle swarm optimizer based on fitness Euclidean-distance ratio. In Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation—GECCO '07, London, UK, 7–11 July 2007; pp. 78–85. [[CrossRef](#)]
49. Li, X. Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization. In Proceedings of the Genetic and Evolutionary Computation Conference 2004, Seattle, WA, USA, 26–30 June 2004; pp. 105–116. [[CrossRef](#)]
50. Islam, J.; Rahaman, A.; Vasant, P.; Negash, B.; Hoqe, A.; Alhitmi, H.K.; Watada, J. A modified niching crow search approach to well placement optimization. *Energies* **2021**, *14*, 857. [[CrossRef](#)]
51. Peram, T.; Veeramachaneni, K.; Mohan, C.K. Fitness-distance-ratio based particle swarm optimization. In Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03, Indianapolis, IN, USA, 24–26 April 2003. [[CrossRef](#)]
52. Bergh, F.V.D.; Engelbrecht, A.P. A study of particle swarm optimization particle trajectories. *Inf. Sci.* **2006**, *176*, 937–971. [[CrossRef](#)]
53. Ray, T.; Saini, P. Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Eng. Optim.* **2001**, *33*, 735–748. [[CrossRef](#)]
54. Deb, K. Optimal design of a welded beam via genetic algorithms. *AIAA J.* **1991**, *29*, 2013–2015. [[CrossRef](#)]
55. Ray, T.; Liew, K. Society and civilization: An optimization algorithm based on the simulation of social behavior. *IEEE Trans. Evol. Comput.* **2003**, *7*, 386–396. [[CrossRef](#)]
56. Hedar, A.-R.; Fukushima, M. Derivative-free filter simulated annealing method for constrained continuous global optimization. *J. Glob. Optim.* **2006**, *35*, 521–549. [[CrossRef](#)]
57. Wang, Y.; Cai, Z.; Zhou, Y. Accelerating adaptive trade-off model using shrinking space technique for constrained evolutionary optimization. *Int. J. Numer. Methods Eng.* **2009**, *77*, 1501–1534. [[CrossRef](#)]
58. Wang, Y.; Cai, Z.; Zhou, Y.; Fan, Z. Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Struct. Multidiscip. Optim.* **2008**, *37*, 395–413. [[CrossRef](#)]
59. Coello, C.A.C.; Montes, E.M. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Adv. Eng. Inform.* **2002**, *16*, 193–203. [[CrossRef](#)]
60. He, Q.; Wang, L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Eng. Appl. Artif. Intell.* **2007**, *20*, 89–99. [[CrossRef](#)]
61. Baykasoğlu, A.; Ozsoydan, F.B. Adaptive firefly algorithm with chaos for mechanical design optimization problems. *Appl. Soft Comput. J.* **2015**, *36*, 152–164. [[CrossRef](#)]

62. Kaveh, A.; Dadras, A. A novel meta-heuristic optimization algorithm: Thermal exchange optimization. *Adv. Eng. Softw.* **2017**, *110*, 69–84. [[CrossRef](#)]
63. Kumar, V.; Kumar, D. An astrophysics-inspired Grey Wolf algorithm for numerical optimization and its application to engineering design problems. *Adv. Eng. Softw.* **2017**, *112*, 231–254. [[CrossRef](#)]
64. Zhang, M.; Luo, W.; Wang, X. Differential evolution with dynamic stochastic selection for constrained optimization. *Inf. Sci.* **2008**, *178*, 3043–3074. [[CrossRef](#)]
65. Eskandar, H.; Sadollah, A.; Bahreininejad, A.; Hamdi, M. Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct.* **2012**, *110–111*, 151–166. [[CrossRef](#)]
66. Sadollah, A.; Bahreininejad, A.; Eskandar, H.; Hamdi, M. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Appl. Soft Comput.* **2013**, *13*, 2592–2612. [[CrossRef](#)]
67. 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](#)]
68. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl. Based Syst.* **2016**, *96*, 120–133. [[CrossRef](#)]