KSÜ Mühendislik Bilimleri Dergisi, 26(2), 2023
Araştırma Makalesi

KSU J Eng Sci, 26(2), 2023
Research Article

Kahramanmaras Sutcu Imam University
Journal of Engineering Sciences

Geliş Tarihi : 02.12.2022
Kabul Tarihi :04.03.2023

Received Date : 02.12.2022
Accepted Date : 04.03.2023

COMBINING GREY WOLF OPTIMIZATION AND WHALE OPTIMIZATION ALGORITHM FOR BENCHMARK TEST FUNCTIONS

KIYASLAMA TEST FONKSİYONLARI İÇİN GRİ KURT OPTİMİZASYONU İLE BALİNA OPTİMİZASYON ALGORİTMASININ BİRLEŞTİRİLMESİ

Mustafa Serter UZER[1] (ORCID: 0000-0002-8829-5987)
Onur INAN[,2*] (ORCID: 0000-0003-4573-7025)

[1] Selçuk Üniversitesi, Elektronik ve Otomasyon Bölümü, Konya, Türkiye
[2] Selçuk Üniversitesi, Bilgisayar Mühendisliği Bölümü, Konya, Türkiye

*Sorumlu Yazar / Corresponding Author: Mustafa Serter UZER, msuzer@selcuk.edu.tr

ABSTRACT

Many optimization problems have been successfully addressed using metaheuristic approaches. These approaches are frequently able to choose the best answer fast and effectively. Recently, the use of swarm-based optimization algorithms, a kind of metaheuristic approach, has become more common. In this study, a hybrid swarm-based optimization method called WOAGWO is proposed by combining the Whale Optimization Algorithm (WOA) and Grey Wolf Optimization (GWO). This method aims to realize a more effective hybrid algorithm by using the positive aspects of the two algorithms. 23 benchmark test functions were utilized to assess the WOAGWO. By running the proposed approach 30 times, the mean fitness and standard deviation values were computed. These results were compared to WOA, GWO, Ant Lion Optimization algorithm (ALO), Particle Swarm Optimization (PSO), and Improved ALO (IALO) in the literature. The WOAGWO algorithm, when compared to these algorithms in the literature, produced the optimal results in 5 of 7 unimodal benchmark functions, 4 of 6 multimodal benchmark functions, and 9 of 10 fixed-dimension multimodal benchmark functions. Therefore, the suggested approach generally outperforms the findings in the literature. The proposed WOAGWO seems to be promising and it has a wide range of uses.

Keywords: Benchmark test functions, whale optimization algorithm, grey wolf optimization.

ÖZET

Bir çok optimizasyon problemi, metasezgisel yaklaşımlar kullanılarak başarıyla ele alınmıştır. Bu yaklaşımlar sıklıkla en iyi yanıtı hızlı ve etkili bir şekilde seçebilmektedir. Son zamanlarda, metasezgisel yaklaşımların bir türü olan sürü tabanlı optimizasyon algoritmalarının kullanımı daha yaygın hale gelmiştir. Bu çalışmada, Balina Optimizasyon Algoritması (WOA) ve Gri Kurt Optimizasyonu (GWO) birleştirilerek WOAGWO adı verilen hibrit sürü tabanlı bir optimizasyon yöntemi önerilmiştir. Bu yöntem, iki algoritmanın olumlu yönlerini kullanarak daha etkin bir hibrit algoritma gerçekleştirmeyi amaçlamaktadır. WOAGWO'yu değerlendirmek için 23 kıyaslama testi işlevi kullanıldı. Önerilen yaklaşım 30 kez çalıştırılarak ortalama uygunluk ve standart sapma değerleri hesaplanmıştır. Bu sonuçlar literatürdeki WOA, GWO, Karınca Aslanı Optimizasyonu algoritması (ALO), Parçacık Sürü Optimizasyonu (PSO) ve Geliştirilmiş ALO (IALO) ile karşılaştırıldı. WOAGWO algoritması, literatürdeki bu algoritmalarla karşılaştırıldığında, 7 unimodal kıyaslama fonksiyonundan 5'inde, 6 multimodal kıyaslama fonksiyonundan 4'ünde ve 10 sabit boyutlu multimodal kıyaslama fonksiyonundan 9'unda en uygun sonuçları vermiştir. Bu nedenle, önerilen yaklaşım genel olarak literatürdeki bulgulardan daha iyi performans göstermektedir. Önerilen WOAGWO ümit verici görünmektedir ve geniş bir kullanım alanına sahiptir.

Anahtar Kelimeler: Balina optimizasyon algoritması, gri kurt optimizasyonu, kıyaslama test fonksiyonları

ToCite: UZER, M.S., & INAN, O., (2023). COMBINING GREY WOLF OPTIMIZATION AND WHALE OPTIMIZATION ALGORITHM FOR BENCHMARK TEST FUNCTIONS. *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi*, 26(2), 462-475.

INTRODUCTION

Typically, the goal of optimization strategies is to narrow down a list of potential options in the issue search space to the best possible solutions. Optimization has drawn a lot of interest in recent years from a variety of disciplines, including engineering, computer science, business, and energy (Mostafa et al., 2022; Oliva & Elaziz, 2020). Static, dynamic, multi-objective, single-objective, constrained, and unconstrained categories of optimization problems can be used to classify optimization issues (Hussien & Amin, 2022). Some classes of problems are difficult to solve using traditional mathematical programming methodologies because numerous topics in artificial intelligence and machine learning are limited, discrete, continuous, uncontrolled, etc. (Abbassi et al., 2019; Faris et al., 2019; Heidari et al., 2019). Developed metaheuristic algorithms (MA) have been employed for a variety of situations as competing problem solvers because of their flexibility, simplicity in usage, and problem-solving ability. These algorithms have the capacity to quickly and effectively identify the ideal solution by utilizing key information from the search space (Arora et al., 2020). MA can be divided into three groups. These are swarm-based, physics-based, and evolution-based methods (Mirjalili & Lewis, 2016). From the literature, the following are a few examples of swarm-based optimizations: Grey Wolf Optimization (GWO) (Mirjalili et al., 2014), Artificial Bee Colony (ABC) (Karaboga & Basturk, 2007; Karaboga & Basturk, 2008), Butterfly Optimization Algorithm(BOA) (Arora & Anand, 2019), Harris Hawks Optimization (HHO) (Heidari et al., 2019), Honey Badger Algorithm (HBA) (Hashim et al., 2022), Grasshopper Optimization Algorithm (GOA) (Mirjalili et al., 2018), Selfish Herd Optimizer (SHO) (Fausto et al., 2017), Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995), Snake Optimizer (SO) (Hashim & Hussien, 2022), Optimal Foraging Algorithm (OFA) (Zhu & Zhang, 2017), Whale Optimization Algorithm (WOA) (Mirjalili & Lewis, 2016).

The population of interacting agents or swarms that have the capacity to self-organize is modeled by the research field known as swarm intelligence (Karaboga & Basturk, 2007). Some of the examples of swarm intelligence or swarm-based optimization are described in the remainder of this section. An example of swarm intelligence is the way bees cluster around their hives. The ABC technique presented by Karaboga and Basturk is based on the intelligent behavior of a honey bee swarm. In this article, ABC algorithm is used for multivariate optimization and compared with other algorithms (Karaboga & Basturk, 2007). The HHO algorithm is presented by Heidari et al.(Heidari et al., 2019). With the help of collaboration between many hawks to attack from varying angles to confuse their prey, the HHO algorithm is simulated and it is based on this swarm scenario. In response to the escape strategies of their prey, hawks have evolved many hunting techniques. In order to create HHO, these distinct tactics were mathematically modeled. In 2016, Mirjalili and Lewis introduced the WOA (Mirjalili & Lewis, 2016). WOA is based on the strategies used by humpback whales, which produce bubbles and constrict their prey into a bubble spiral. Humpback whales gather their food, which frequently consists of fish communities, with the aid of air bubbles. They then reach the surface of the sea and wrap the target fishes in a more restricted region. The stages of WOA are prey encircling, a bubble-net attack unleashing, and prey searching. To estimate daily reference evapotranspiration using meteorological data, a hybrid technique that combines extreme gradient boosting (XGB) and WOA was developed in (Yan et al., 2021). Two binary WOA algorithm variants are presented to identify the optimal feature subsets in (Mafarja & Mirjalili, 2018). Mirjalili et al. created GWO, an optimization system that simulates the hunting techniques used by grey wolves in the wild (Mirjalili et al., 2014). It is established on group hunting strategies that grey wolves used to hunt in accordance with their social structures. Moreover, many problems of GWO algorithm have been applied and some of them are as follows. For multi-level thresholding segmentation, the discrete multi-objective shuffled gray wolf optimizer (D-MOSG) technique was presented in (Karakoyun et al., 2021). To find a solution to the multi-objective optimization problems, multi-objective shuffled GWO (MOSG) has been proposed in (Karakoyun et al., 2020). An improved form of the Ant Lion Optimization algorithm (IALO) has been proposed in (Toz, 2019). 23 benchmark functions and image clustering problems have been solved with IALO. In cases where optimization algorithms alone are not sufficient, researchers have tried hybridizing the algorithms. In optimization problems, hybrid optimization methods can sometimes give better results than using a single optimization problem. In order to provide solutions for scheduling challenges in cloud jobs, a hybrid method called HGWWO is presented in (Ababneh, 2021). This method is established using GWO and WOA. A novel Hybrid WOA with gathering strategies (HWOAG) has been proposed for solving high-dimensional problems in (Zhang & Wen, 2021).

The main contributions of this article can be summarized as follows:

- In this research, a hybrid optimization technique called WOAGWO was created by combining the WOA and GWO algorithms to achieve improved optimization outcomes.

- 23 benchmark test functions have been used to evaluate this method's performance.
- After running the suggested method 30 times, average fitness and standard deviation values were acquired.
- The WOAGWO algorithm's findings were contrasted with those of other techniques found in literature and the proposed algorithm was found to perform better than previously reported techniques.
- According to the literature has shown that the suggested method is promising and that it may be used in a variety of applications.

The remainder of the article has covered the WOA and GWO algorithms in the 'Methods' section. The proposed hybrid WOAGWO algorithm is detailed in the section called 'The Proposed Method', while in the next section, the results of the proposed algorithm, discussion, and their comparison with the literature are presented. The article concludes with a summary of the findings of the WOAGWO algorithm.

METHODS

Whale Optimization Algorithm (WOA)

Due to their ability to generate answers for issues in a variety of sectors, swarm-based optimization algorithms have gained popularity recently. Mirjalili and Lewis created the WOA algorithm in 2016, drawing inspiration from how humpback whales compress their prey into the bubble spiral they create while foraging (Mirjalili & Lewis, 2016).

The hunting strategy of humpback whales involves creating air bubbles to corral groups of small fish, then narrowing their focus to target fish. The three stages of WOA include enclosing prey, using a bubble-net attack, and looking for prey.

Humpback whales locate their target during the encircling prey stage, at which point they wrap their target. In this method, the position of the prey is the ideal outcome. The best solution value found to get to this optimum value is updated for other solutions. This behavior is mathematically predicted by Equations (1) and (2).

$$\vec{D} = \left| \vec{C}.\vec{X}^*(t) - \vec{X}(t) \right| \quad (1)$$

$$\vec{X}(t+1) = \left| \vec{X}^*(t) - \vec{A}.\vec{D} \right| \quad (2)$$

where X and X^* denote the position vector, and the best solution position vector, respectively. Also, t refers to the iteration currently in progress.

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \quad (3)$$

$$\vec{C} = 2.\vec{r} \quad (4)$$

With the help of r, a random vector between 0 and 1, and a, a linearly decreasing value from 2 to 0, the vectors A and C, which are coefficient vectors in Equations (3) and (4), are created. The following formula can be used to derive the a in Equation (3):

$$a = 2 - t\frac{2}{MaxIter} \quad (5)$$

The number of iterations is given in Equation (5) as t, and the maximum number is denoted as $MaxIter$. Before creating the spiral formula between the current solution and the best leading solution, the distance between the solutions of X and X^* is used to compute the path that has a spiral shape. In this case, X is the answer, and X^* is the leading answer. By lowering a in Equation (3), the diminishing encircling movement occurs. Equation (6) describes the spiral movement.

$$\vec{X}(t+1) = D'.e^{bl}.\cos(2\pi l) + \vec{X}^*(t) \tag{6}$$

In Equation (6), b denotes shape of the spiral, $\vec{D}'$ is the distance between a whale X and a prey, l is a random number between -1 and 1(Mafarja & Mirjalili, 2018).

Equation (7) represents the situation in which whales choose one of these two-shaped paths at random.

$$\vec{X}(t+1) = \begin{cases} Use\ Eq.2 & if\ (p < 0.5) \\ Use\ Eq.6 & if\ (p \geq 0.5) \end{cases} \tag{7}$$

During the search for prey stage, the whale normally searches for prey at random. As a result, vector A is used, which generates random values. Equations (8) and (9) express this mechanism (Mafarja & Mirjalili, 2018).

```
Create initial whale' population Xi (i = 1, 2, ..., n)
Compute fitness values for each solution
X*=the best solution
while (t < Max_Iteration)
    for each solution
        Update a, A, C, l, and p
        If1 (p<0.5)
                if2 (|A| < 1)
                        Update the location of the current solution by computing Eq. (2).
                else if2 (|A| >=1)
                        Choose a solution (x_rand) at random.
                        Calculate Eq. (9)
                end if2
        else if1 (p >0.5)
                Update the current search's location using the Eq. (6)
        end if1
    end for
        Verify and correct any solutions that extend beyond the search space.
        Compute each solution's fitness and Update X* if a better solution emerges.
        t=t+1
end while
return X*
```

Figure 1. Pseudocode of the WOA Algorithm

$$\vec{D} = \left| \vec{C}.\vec{X}_{rand} - \vec{X} \right| \tag{8}$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A}.\vec{D} \tag{9}$$

In the case of $|A|<1$, Equation (2) is used, and in the case of $|A|>=1$, Equation (9) is used. The WOA algorithm pseudocode is shown in Figure 1 (Mirjalili & Lewis, 2016).

Grey Wolf Algorıthm (GWO)

The GWO method was formulated by Mirjalili et al. and is inspired by the hunting strategies employed by grey wolves in their social hierarchy (Mirjalili et al., 2014). The grey wolves are arranged in four groups, as illustrated in Figure 2: alpha (α), beta (β), delta (δ) and omega (ω). The alpha group, which consists of a male and a female wolf, is responsible for making important decisions such as hunting. The beta wolves assist the alpha wolves in implementing their decisions. The delta wolves occupy the third position in the pack hierarchy and are subservient to the alpha and beta wolves. The omega wolves are last in line to be allowed to eat, as they are at the bottom of the wolf hierarchy.

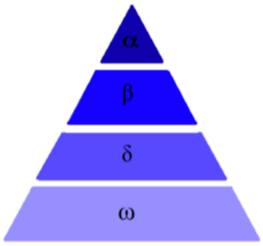

Figure 2. Hierarchy of Grey Wolves

The GWO algorithm has four stages: surrounding the prey, hunting, assaulting the prey (exploitation), and seeking prey (exploration). The grey wolves surrounding the prey are expressed by Equations (10) and (11).

$$\vec{H}(t+1)=\vec{H}l(t)-\vec{A}.\vec{D} \tag{10}$$

$$\vec{D}=\left|\vec{C}.\vec{H}l(t)-\vec{H}(t)\right| \tag{11}$$

where Equations (12) and (13) produce *A* and *C* coefficient vectors. The location of the target at the *i* iteration is denoted by $\vec{H}l(t)$, whereas the position of a grey wolf is indicated by $\vec{H}(t+1)$.

$$\vec{A}=2\vec{a}.\vec{r_1}+\vec{a} \tag{12}$$

$$\vec{C} = 2.\vec{r_2} \tag{13}$$

The value of *a* decreases in a linear fashion from 2 to 0. The $\vec{r_1}$ and $\vec{r_2}$ variables are made up of random values between [0, 1]. By altering the *A* and *C* vectors, the best agent's closest new places may be controlled.

Once the encircling stage is complete, the search for the best solution commences during the hunting phase. During this stage, alpha leads the hunt, with beta and delta joining in. As a result, the three best positions are used to regenerate grey wolves' position by Equations (14), (15), and (16).

$$\vec{D}_\alpha=\left|\vec{C_1}.\vec{H}_\alpha-\vec{H}\right|,\ \vec{D}_\beta=\left|\vec{C_2}.\vec{H}_\beta-\vec{H}\right|,\ \vec{D}_\delta=\left|\vec{C_3}.\vec{H}_\delta-\vec{H}\right| \tag{14}$$

$$\vec{H_1}=\vec{H}_\alpha-\vec{A_1}.\vec{D}_\alpha,\ \vec{H_2}=\vec{H}_\beta-\vec{A_2}.\vec{D}_\beta,\ \vec{H_3}=\vec{H}_\delta-\vec{A_3}.\vec{D}_\delta \tag{15}$$

$$\vec{H}(t+1)=\frac{\vec{H_1}+\vec{H_2}+\vec{H_3}}{3} \tag{16}$$

To summarize, the GWO algorithm generates a haphazard population. The position of probable prey is predicted by delta, alpha, and beta wolves. The distance of the possible solution is then updated. Then, from 2 to 0, *a* is lowered to emphasize exploration and exploitation. If *A<1*, they proceed to the prey. If *A>1*, they stop assaulting the prey. GWO is terminated when it has achieved a satisfying conclusion. GWO is depicted in Figure 3 as a pseudocode. (Mirjalili et al., 2014).

```
Create grey wolf' population Hi (i = 1, 2, ..., n)
Create a, A, C
Compute each search agent's fitness.
Hα= The most effective search agent
Hβ= The second most effective search agent
Hδ= The third most effective search agent
  while (t < Max_iteration)
    for every search agent
      Upgrade current search agent position using Equation (16)
    end for
  Upgrade a, A, C
  Compute all search agent's fitness.
  Upgrade Hα, Hβ, Hδ
  t=t+1
  end while
return Hα
```

Figure 3. The GWO Algorithm's Pseudocode

THE PROPOSED METHOD

In addressing comparison functions or engineering challenges, WOA and GWO algorithms may not be adequate to get the desired value. With the method we created, a hybrid algorithm dubbed WOAGWO is suggested, which highlights the best qualities of WOA and GWO. Figure 4 depicts the WOAGWO algorithm's pseudocode.

```
Create random initialize population Xk(k = 1, 2, . . . , N)
r=0, Best_score=10E+10;
while (r<T) do
  [Leader_score, Leader _pos, wPositions]=WOA(Positions, fobj, r);
  if Leader_score < Best_score
  Best_score = Leader_score;
  Best_pos = Leader_pos;
  Positions=wPositions;
  else
  r=r+1;
  [Alpha_score, Alpha_pos, gPositions]=GWO(Positions, fobj, r);
  if Alpha_score < Best_score
  Best_score = Alpha_score;
  Best_pos = Alpha_pos;
  Positions=gPositions;
  end if
  end if
  r=r+1
end while
Return Best_score, Best_pos
```

Figure 4. Pseudocode of the Proposed WOAGWO

After the initial population and initial parameters are established, the fitness function of WOA is first calculated in the proposed WOAGWO technique. If WOA's leader score value is better than the best score value, the WOA leader score value is assigned as the best score value. In addition, the WOA leader position value is assigned as the best position value and the population values of WOA are assigned as the best population value. If WOA's leader score value is worse than the best score value, the GWO Alpha score value is assigned as the best score value. In addition, the GWO Alpha position value is assigned as the best position value and the population values of GWO are assigned as the best population value. This procedure is repeated until the desired maximum iteration is reached.

RESULTS AND DISCUSSION

First, to measure the success of the proposed WOAGWO algorithm, the results of 23 benchmark test functions were found. Of these benchmark test functions, those between F1 and F7 are called single-mode, those between F8-F13 are called multimodal, and those between F14-F23 are called fixed-size multimodal functions. The results obtained from these functions were then used to compare with different methods in the literature. In order to make this literature comparison more reliable, the same parameters were used as much as possible and it was tried to reach the articles containing the same parameters. In the literature, the number of independent runs, the number of population, and the maximum number of iterations for WOA, GWO, and PSO algorithms have been used as 30, 30, and 500, respectively (Mirjalili & Lewis, 2016; Mirjalili et al., 2014). The number of independent runs, the number of population, and the maximum number of iterations for the ALO and IALO algorithms have been utilized in the literature as 30, 40, and 500, respectively (Toz, 2019). To compare with the literature, independent runs, the number of population and the maximum number of iterations for the proposed WOAGWO algorithm have been chosen as 30, 30, and 500, respectively. The mean and standard deviation values were calculated using these 30 times run results. When the proposed WOAGWO and GWO, WOA, ALO, PSO, and IALO algorithms were compared, in general, WOAGWO was discovered that better results were attained. Table 1 shows the unimodal F1-F7 functions (Mirjalili & Lewis, 2016). The function's limitations in the search space are represented by *Range*, while the optimum value is represented by F_{min}. The function dimension is denoted by *Dim*. The F1-F7 functions can measure the performance and capabilities of the optimization method during use.

Table 1 Unimodal Functions

Func. Num	Function	F_{min}	*Range* Lb	Ub	*Dim*
1	$\sum_{i=1}^{n} d_i^2$	0	-100	100	30
2	$\sum_{i=1}^{n} \lvert d_i \rvert + \prod_{i=1}^{n} \lvert d_i \rvert$	0	-10	10	30
3	$\sum_{i=1}^{n} (\sum_{j-1}^{i} d_j)^2$	0	-100	100	30
4	$\max_i \{\lvert d_i \rvert, 1 \le i \le n\}$	0	-100	100	30
5	$\sum_{i=1}^{n-1} [100(d_{i+1} - d_i^2)^2 + (d_i - 1)^2]$	0	-30	30	30
6	$\sum_{i=1}^{n} ([d_i + 0.5])^2$	0	-100	100	30
7	$\sum_{i=1}^{n} i d_i^4 + random[0,1)$	0	-1.28	1.28	30

Table 2. Results of F1-F7 Functions

Functions		1	2	3	4	5	6	7	Winner/Total
GWO(Mirjalili et al., 2014)	Ave.	6.5900E-28	7.1800E-17	3.2900E-06	**5.6100E-07**	2.6813E+01	8.1658E-01	2.2130E-03	1/7
	Std.	6.3400E-05	2.9014E-02	7.9150E+01	1.3151E+00	6.9905E+01	1.2600E-04	1.0029E-01	
WOA(Mirjalili & Lewis, 2016)	Ave.	1.4100E-30	1.0600E-21	5.3900E-07	7.2581E-02	2.7866E+01	3.1163E+00	1.4250E-03	0/7
	Std.	4.9100E-30	2.3900E-21	2.9300E-06	3.9747E-01	7.6363E-01	5.3243E-01	1.1490E-03	
ALO (Toz, 2019)	Ave.	4.3800E-09	5.5354E-01	6.5900E-04	8.5600E-04	2.7842E+01	4.6200E-09	1.5767E-02	0/7
	Std.	1.8100E-09	1.3245E+00	8.3500E-04	1.1980E-03	6.2201E+01	2.2200E-09	9.8230E-03	
PSO(Mirjalili & Lewis, 2016)	Ave.	1.3600E-04	4.2144E-02	7.0126E+01	1.0865E+00	9.6718E+01	1.0200E-04	1.2285E-01	0/7
	Std.	2.0200E-04	4.5421E-02	2.2119E+01	3.1704E-01	6.0116E+01	8.2800E-05	4.4957E-02	
IALO (Toz, 2019)	Ave.	3.6800E-11	3.4600E-04	5.7663E-01	2.7898E-02	3.4984E+02	**4.5400E-11**	1.3740E-02	1/7
	Std.	1.1400E-10	7.7400E-04	6.2807E-01	9.2275E-02	7.4489E+02	1.7600E-10	9.3790E-03	
WOAGWO	Ave.	**6.9361E-63**	**4.3941E-43**	**1.3401E-07**	2.2606E-03	**2.6575E+01**	3.4826E-01	**1.2430E-03**	**5/7**
	Std.	2.6601E-62	2.3711E-42	2.6048E-07	8.3332E-03	8.4498E-01	3.1568E-01	4.6156E-04	

Table 2 displays the results of the F1-F7 functions. WOAGWO algorithm obtained the optimal result for 5 of 7 unimodal comparison functions. In other words, it has been seen that it is better than other comparison algorithms except for F4 and F6 results. The GWO algorithm for F4 and the IALO algorithm for F6 achieved the best results. In light of the results obtained, it is seen that the proposed WOAGWO algorithm provides an improvement in 7 functions according to the WOA algorithm, while an improvement is achieved in 6 functions according to the GWO algorithm. Figure 5 shows the best representations of F1-F7 functions in search space and objective space. The F5 and F7 functions reach the optimal solution before the maximum number of iterations is reached. Conversely, other functions converge the target value when they are very close to the maximum iteration value.

In order to arrive at the global optimum, an optimization algorithm seeks to avoid all local optima. This operation can be tested using the F8-F13 functions. In essence, these benchmark functions can assess the performance of an optimization algorithm in terms of its ability to evade local optima. Table 3 displays F8-F13 functions (Mirjalili & Lewis, 2016).

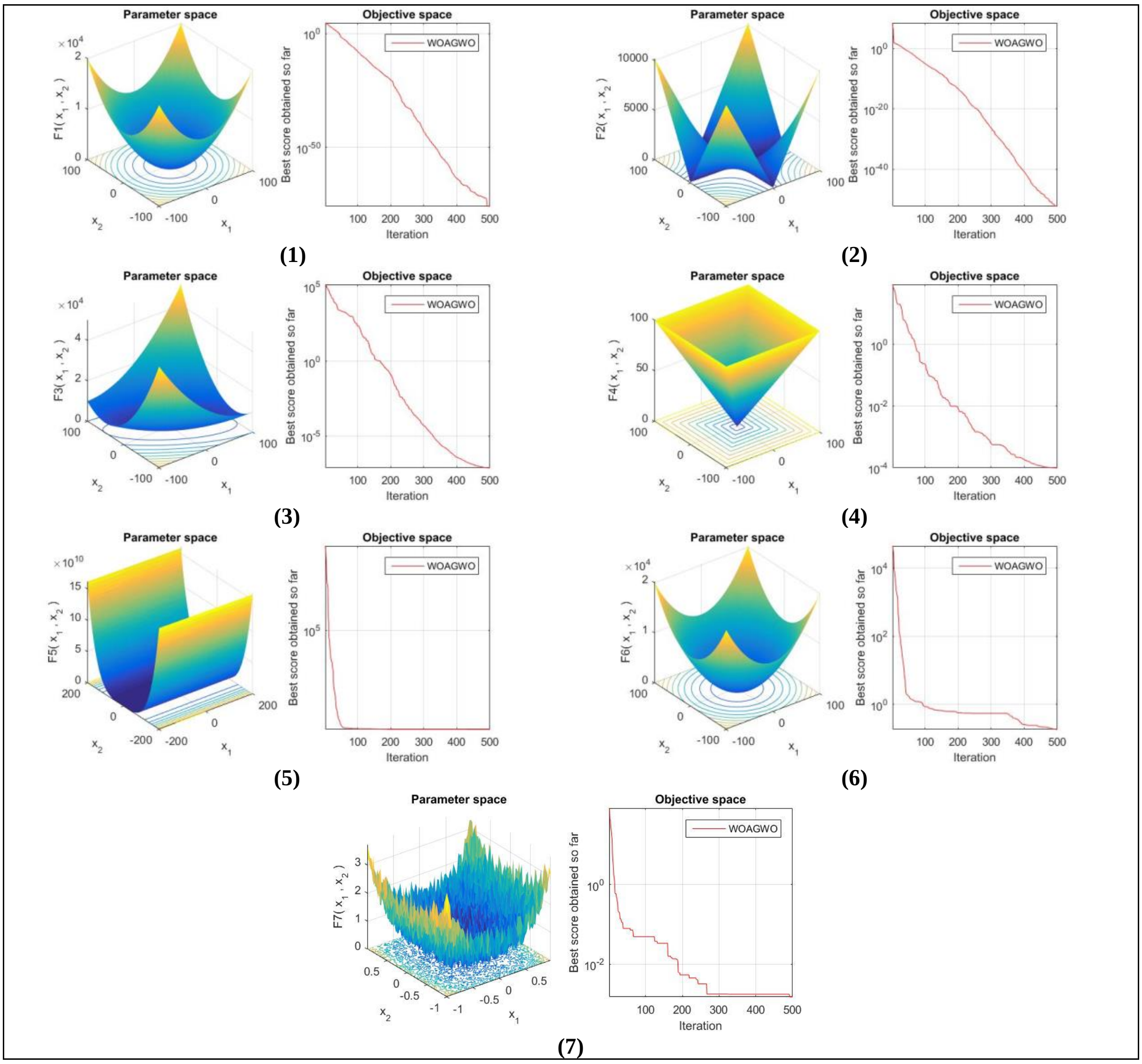

Figure 5. The Best Representations of F1-F7 Functions in Search Space and Objective Space

Table 4 displays the F8-F13 functions' results (Mirjalili & Lewis, 2016). The WOAGWO algorithm obtained the optimal result for 4 of 6 multimodal comparison functions. In other words, it has been seen that it is better than different comparison algorithms except for F8 and F13 results. For F13 and F8, the ALO algorithm achieved the best results. It is seen that the proposed WOAGWO algorithm gives the same or better optimum results in 5 of 6 functions according to the WOA algorithm, and the same or better optimum results in 5 of 6 functions according to the GWO algorithm.

Figure 6 shows the best representations of F8-F13 functions in search space and objective space. The F8 and F10 functions converge the target value before reaching the maximum iteration value. Conversely, F9, F11, F12, and F13 functions converge the target value when they are very close to the maximum iteration value.

Table 3 Multimodal Functions

Func. Num	Function	F_{min}	Range Lb	Ub	Dim
8	$\sum_{i=1}^{n} -d_i \sin(\sqrt{\lvert d_i \rvert})$	-418.9829x5	-500	500	30
9	$\sum_{i=1}^{n} \left[d_i^2 - 10\cos(2\pi d_i + 10) \right]$	0	-5.12	5.12	30
10	$-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} d_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi d_i)) + 20 + e$	0	-32	32	30
11	$\frac{1}{4000}\sum_{i=1}^{n} d_i^2 - \prod_{i=1}^{n}\cos(\frac{d_i}{\sqrt{i}}) + 1$	0	-600	600	30
12	$\frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(d_i, 10, 100, 4)$ $y_i = 1 + \frac{d_i + 1}{4} \quad u(d_i, a, k, m) = \begin{cases} k(d_i - a)^m & d_i > a \\ 0 & -a < d_i < a \\ k(-d_i - a)^m & d_i < -a \end{cases}$	0	-50	50	30
13	$0.1\left\{\sin^2(3\pi d_1) + \sum_{i=1}^{n}(d_i - 1)^2\left[1 + \sin^2(3\pi d_i + 1)\right] + (d_n - 1)^2\left[1 + \sin^2(2\pi d_n)\right]\right\} + \sum_{i=1}^{n} u(d_i, 5, 100, 4)$	0	-50	50	30

Table 4. Results of F8-F13 Functions

Functions		8	9	10	11	12	13	Winner/Total
GWO(Mirjalili et al., 2014)	Ave.	-6.1231E+03	3.1052E-01	1.0600E-13	4.4850E-03	5.3438E-02	6.5446E-01	0/6
	Std.	-4.0874E+03	4.7356E+01	7.7835E-02	6.6590E-03	2.0734E-02	4.4740E-03	
WOA(Mirjalili & Lewis, 2016)	Ave.	-5.0808E+03	**0.0000E+00**	7.4043E+00	2.8900E-04	3.3968E-01	1.8890E+00	1/6
	Std.	6.9580E+02	0.0000E+00	9.8976E+00	1.5860E-03	2.1486E-01	2.6609E-01	
ALO (Toz, 2019)	Ave.	**-2.4391E+03**	1.9402E+01	2.9240E-01	2.2125E-01	1.4850E+00	**7.0100E-04**	2/6
	Std.	4.4985E+02	1.1247E+01	6.1341E-01	1.0754E-01	1.7889E+00	3.8380E-03	
PSO(Mirjalili & Lewis, 2016)	Ave.	-4.8413E+03	4.6704E+01	2.7602E-01	9.2150E-03	6.9170E-03	6.6750E-03	0/6
	Std.	1.1528E+03	1.1629E+01	5.0901E-01	7.7240E-03	2.6301E-02	8.9070E-03	
IALO (Toz, 2019)	Ave.	-2.8191E+03	1.4725E+01	7.8411E-01	2.0489E-01	1.1932E-01	2.1990E-03	0/6
	Std.	3.1346E+02	5.0693E+00	1.0061E+00	1.0017E-01	2.3377E-01	4.4720E-03	
WOAGWO	Ave.	-8.1152E+03	**0.0000E+00**	**8.8818E-16**	**0.0000E+00**	**4.4121E-03**	5.6975E-01	**4/6**
	Std.	4.7598E+02	0.0000E+00	0.0000E+00	0.0000E+00	3.2210E-03	2.3965 E-01	

Fixed-dimension multimodal functions from F14 to F23 are depicted in Table 5 (Mirjalili & Lewis, 2016). These functions use to evaluate how well exploitation and exploration of the search process are balanced in optimization algorithms.

The results obtained from the first 5 and the last 5 of the F14-F23 functions are given in Table 6 and Table 7, respectively. The WOAGWO algorithm achieved the optimal result in 9 of 10 fixed-dimension multimodal comparison functions. It has been seen that the proposed WOAGWO algorithm is better than other comparison algorithms except for the F17 result only. The WOA algorithm for F17 achieved the best results. It is seen that the proposed WOAGWO algorithm gives the same or better optimum results in 9 of 10 functions according to the WOA algorithm, and the same or better optimum results in 10 of 10 functions according to the GWO algorithm. In Figure 7, the best representations of F14-F23 functions in search space and objective space are given. Except for F15, other functions in this group converge to the target values before reaching the maximum iteration value.

Comparing the WOAGWO to WOA, GWO, ALO, PSO, and IALO algorithms in the literature, we found that WOAGWO performed optimally in 5 of 7 unimodal benchmark functions, 4 of 6 multimodal benchmark functions, and 9 of 10 fixed-dimension multimodal benchmark functions. In total, 18 of 23 benchmark functions have been found the results closest to the target value. The winner/total ratios in Tables 2, 4, and 7 are given for better comparison. These ratios are obtained by increasing the winner value by 1 when the benchmark function results are higher or the same compared to other algorithms. According to all results of the proposed approach, it generally performs better than the findings in the literature.

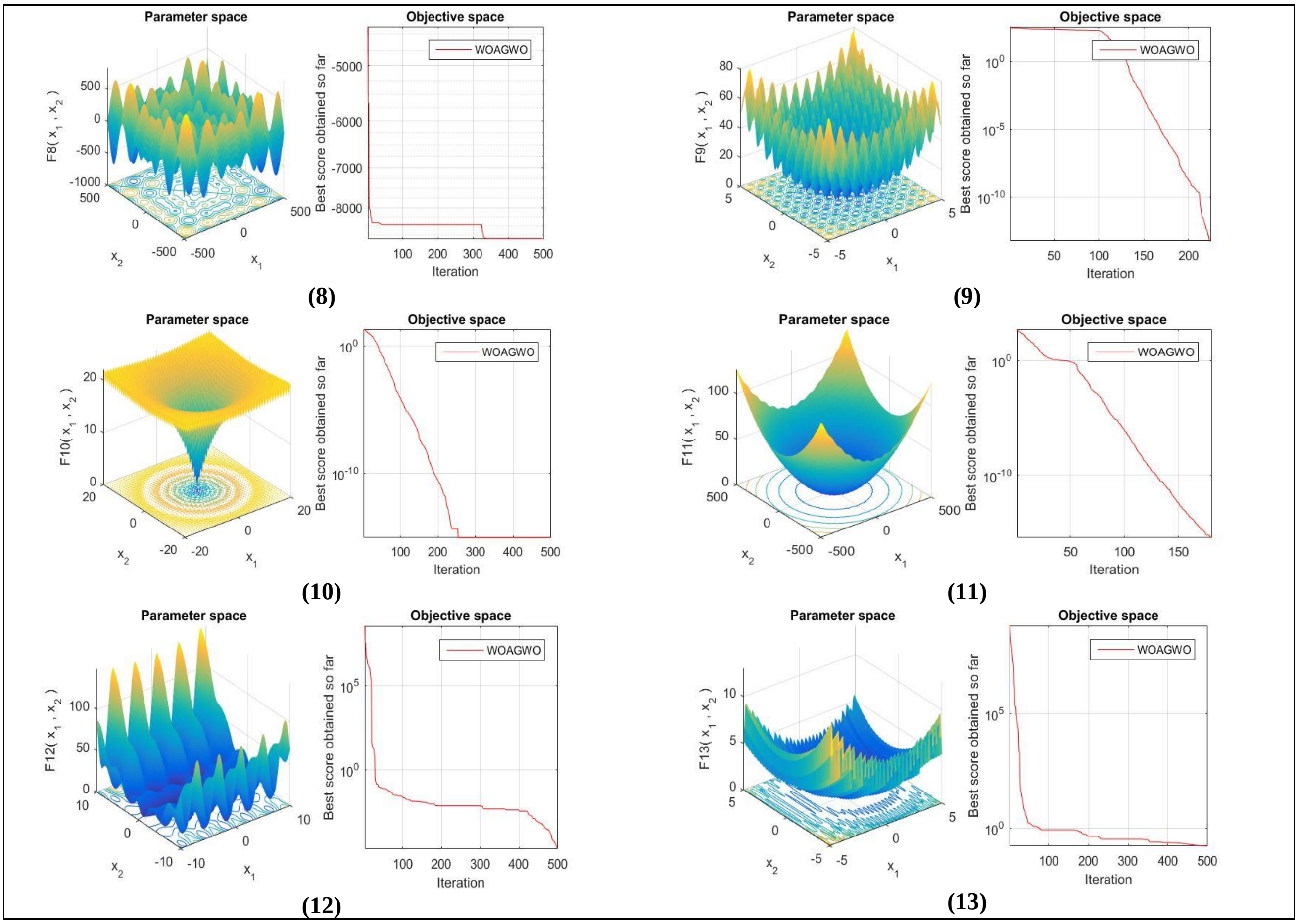

Figure 6. The Best Representations of F8-F13 Functions in Search Space and Objective Space

Table 5. Fixed-Dimension Multimodal Functions

Func. Num	Function	F_{min}	*Range* Lb	Ub	*Dim*
14	$(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(d_i-a_{ij})^6})^{-1}$	1	-65	65	2
15	$\sum_{i=1}^{11}\left[a_i-\frac{d_1(b_i^2+b_id_2)}{b_i^2+b_id_3+d_4}\right]^2$	0.00030	-5	5	4
16	$4d_1^2-2.1d_1^4+\frac{1}{3}d_1^6+d_1d_2-4d_2^2+4d_2^4$	-1.0316	-5	5	2
17	$(d_2-\frac{5.1}{4\pi^2}d_1^2+\frac{5}{\pi}d_1-6)^2+10(1-\frac{1}{8\pi})\cos d_1+10$	0.398	-5	5	2
18	$\left[1+(d_1+d_2+1)^2(19-14d_1+3d_1^2-14d_2+6d_1d_2+3d_2^2)\right]\times \left[30+(2d_1-3d_2)^2\times(18-32d_1+12d_1^2+48d_2-36d_1d_2+27d_2^2)\right]$	3	-2	2	2
19	$-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{3}a_{ij}(d_j-p_{ij})^2)$	-3.86	1	3	3
20	$-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{6}a_{ij}(d_j-p_{ij})^2)$	-3.32	0	1	6
21	$-\sum_{i=1}^{5}\left[(D-a_i)(D-a_i)^T+c_i\right]^{-1}$	-10.1532	0	10	4
22	$-\sum_{i=1}^{7}\left[(D-a_i)(D-a_i)^T+c_i\right]^{-1}$	-10.4028	0	10	4
23	$-\sum_{i=1}^{10}\left[(D-a_i)(D-a_i)^T+c_i\right]^{-1}$	-10.5363	0	10	4

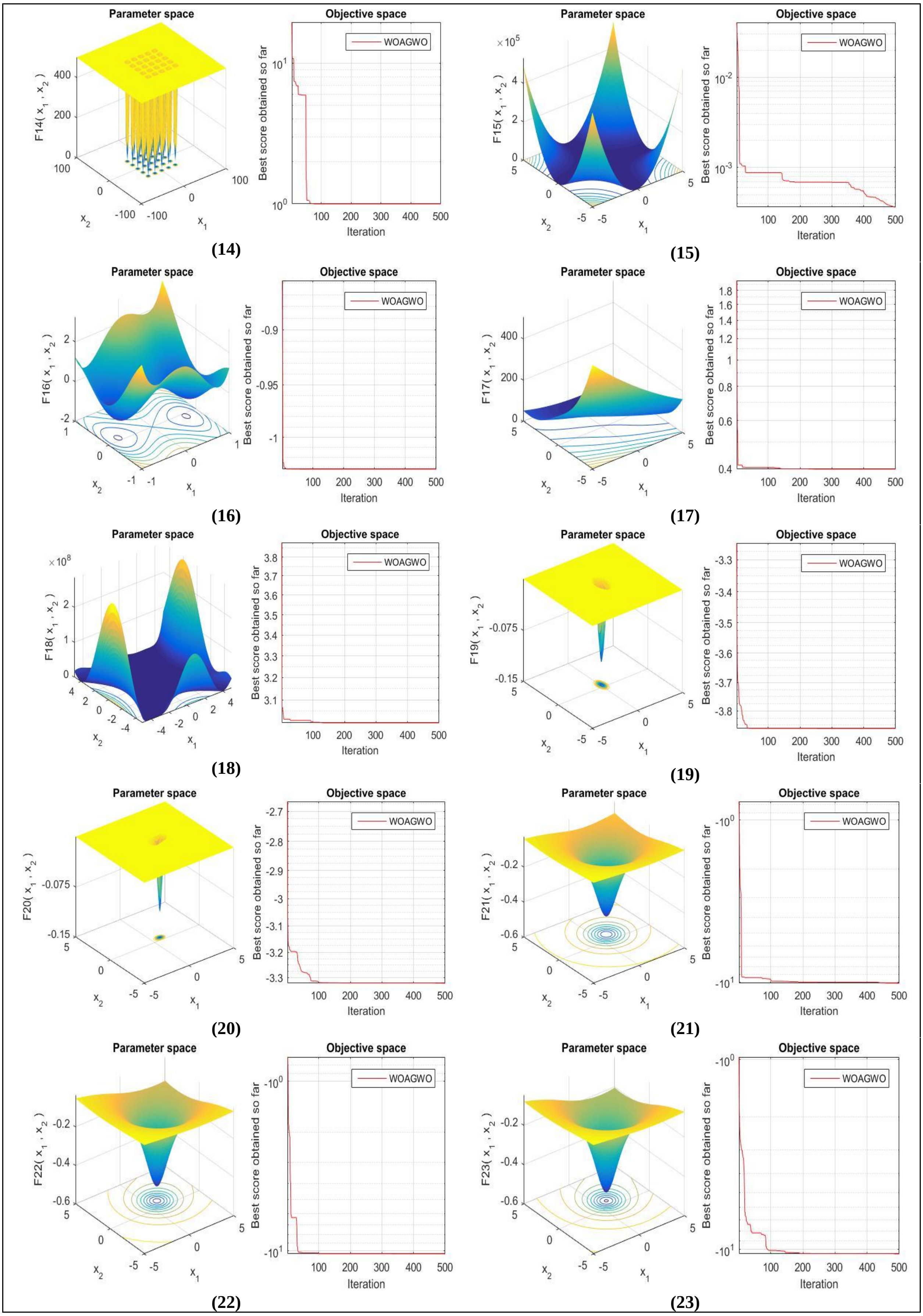

Figure 7. The Best Representations of F14-F23 Functions in Search Space and Objective Space

Table 6. Results of the First 5 of the F14-F23 Functions

Functions		14	15	16	17	18
GWO(Mirjalili et al., 2014)	Ave.	4.0425E+00	3.3700E-04	**-1.0316E+00**	3.9789E-01	**3.0000E+00**
	Std.	4.2528E+00	6.2500E-04	-1.0316E+00	3.9789E-01	3.0000E+00
WOA(Mirjalili & Lewis, 2016)	Ave.	2.1120E+00	5.7200E-04	**-1.0316E+00**	**3.9791E-01**	3.0000E+00
	Std.	2.4986E+00	3.2400E-04	4.2000E-07	2.7000E-05	4.2200E-15
ALO (Toz, 2019)	Ave.	2.7076E+00	2.8430E-03	**-1.0316E+00**	3.9789E-01	**3.0000E+00**
	Std.	2.3599E+00	5.9450E-03	9.8600E-14	5.5900E-14	3.3100E-13
PSO(Mirjalili & Lewis, 2016)	Ave.	3.6272E+00	5.7700E-04	-1.0316E+00	3.9789E-01	3.0000E+00
	Std.	2.5608E+00	2.2200E-04	6.2500E-16	0.0000E+00	1.3300E-15
IALO (Toz, 2019)	Ave.	1.2295E+00	2.1910E-03	**-1.0316E+00**	3.9789E-01	**3.0000E+00**
	Std.	6.2052E-01	4.9450E-03	5.7600E-16	0.0000E+00	4.9300E-15
WOAGWO	Ave.	**1.1965E+00**	**3.1143E-04**	**-1.0316E+00**	3.9789E-01	**3.0000E+00**
	Std.	5.4668E-01	7.7603E-06	4.8114E-07	1.6085E-06	9.3406E-06

Table 7. Results of the Last 5 of the F14-F23 Functions

Functions		19	20	21	22	23	Winner/Total
GWO(Mirjalili et al., 2014)	Ave.	-3.8626E+00	-3.2865E+00	**-1.0151E+01**	-1.0402E+01	-1.0534E+01	3/10
	Std.	-3.8628E+00	-3.2506E+00	-9.1402E+00	-8.5844E+00	-8.5590E+00	
WOA(Mirjalili & Lewis, 2016)	Ave.	-3.8562E+00	-2.9811E+00	-7.0492E+00	-8.1818E+00	-9.3424E+00	2/10
	Std.	2.7060E-03	3.7665E-01	3.6296E+00	3.8292E+00	2.4147E+00	
ALO (Toz, 2019)	Ave.	-3.8628E+00	-3.2624E+00	-6.3766E+00	-7.1015E+00	-8.2471E+00	2/10
	Std.	2.3000E-13	6.0657E-02	3.2796E+00	3.4428E+00	3.3601E+00	
PSO(Mirjalili & Lewis, 2016)	Ave.	-3.8628E+00	-3.2663E+00	-6.8651E+00	-8.4565E+00	-9.9529E+00	0/10
	Std.	2.5800E-15	6.0516E-02	3.0196E+00	3.0871E+00	1.7828E+00	
IALO (Toz, 2019)	Ave.	-3.8628E+00	-3.2775E+00	-7.2848E+00	-8.3333E+00	-8.2543E+00	2/10
	Std.	2.8900E-12	5.9564E-02	2.7947E+00	3.2587E+00	3.3636E+00	
WOAGWO	Ave.	**-3.8619E+00**	**-3.2978E+00**	**-1.0151E +01**	**-1.04022E +01**	**-1.05351E +01**	**9/10**
	Std.	2.3817E-03	4.8753 E-02	4.9824E-03	5.4572E-04	2.5352E-03	

CONCLUSION

A very successful and effective hybrid algorithm consisting of the combination of WOA and GWO, which are swarm-based optimizations, is presented. This hybrid algorithm is called WOAGWO. Using 23 benchmark functions, the suggested algorithm's effectiveness was evaluated. These test functions consist of three groups that measure the algorithm's capacity, the algorithm's power to avoid local optimum, and the algorithm's balance between exploitation and exploration. The results obtained from these three test groups show the efficiency of the proposed algorithm. In order to obtain more reliable results, the proposed method was run 30 times, and then calculate the mean fitness value and standard deviation value. Moreover, the best graphs of both search space and objective space obtained as a result of running these 30 times are presented.

The proposed WOAGWO's mean value was compared to that of the GWO, WOA, ALO, PSO, and IALO algorithms used in the literature. When the proposed algorithm and the literature were compared, it was discovered that WOAGWO produced the best results in 9 of 10 fixed-dimension multimodal benchmark functions, 4 of 6 multimodal benchmark functions, and 5 of 7 unimodal benchmark functions. Consequently, the suggested approach generally performs better than the findings in the literature. Therefore, we plan to use it in engineering applications in the future.

REFERENCES

Ababneh, J. (2021). A Hybrid Approach Based on Grey Wolf and Whale Optimization Algorithms for Solving Cloud Task Scheduling Problem. *Mathematical Problems in Engineering, 2021*.

Abbassi, R., Abbassi, A., Heidari, A. A., & Mirjalili, S. (2019). An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models. *Energy Conversion and Management*, *179*, 362-372. https://doi.org/10.1016/j.enconman.2018.10.069

Arora, S., & Anand, P. (2019). Binary butterfly optimization approaches for feature selection. *Expert Systems with Applications*, *116*, 147-160.

Arora, S., Sharma, M., & Anand, P. (2020). A Novel Chaotic Interior Search Algorithm for Global Optimization and Feature Selection. *Applied Artificial Intelligence, 34*(4), 292-328. https://doi.org/10.1080/08839514.2020.1712788

Faris, H., Al-Zoubi, A. M., Heidari, A. A., Aljarah, I., Mafarja, M., Hassonah, M. A., & Fujita, H. (2019). An intelligent system for spam detection and identification of the most relevant features based on evolutionary Random Weight Networks. *Information Fusion, 48*, 67-83. https://doi.org/10.1016/j.inffus.2018.08.002

Fausto, F., Cuevas, E., Valdivia, A., & Gonzalez, A. (2017). A global optimization algorithm inspired in the behavior of selfish herds. *Biosystems, 160*, 39-55. https://doi.org/10.1016/j.biosystems.2017.07.010

Hashim, F. A., Houssein, E. H., Hussain, K., Mabrouk, M. S., & Al-Atabany, W. (2022). Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems. *Mathematics and Computers in Simulation*, *192*, 84-110. https://doi.org/10.1016/j.matcom.2021.08.013

Hashim, F. A., & Hussien, A. G. (2022). Snake Optimizer: A novel meta-heuristic optimization algorithm. *Knowledge-Based Systems*, *242*, 108320.

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. L. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems-the International Journal of Escience*, *97*, 849-872.

Hussien, A. G., & Amin, M. (2022). A self-adaptive Harris Hawks optimization algorithm with opposition-based learning and chaotic local search strategy for global optimization and feature selection. *International Journal of Machine Learning and Cybernetics, 13*(2), 309-336. https://doi.org/10.1007/s13042-021-01326-4

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*(3), 459-471. https://doi.org/10.1007/s10898-007-9149-x

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, *8*(1), 687-697.

Karakoyun, M., Gulcu, S., & Kodaz, H. (2021). D-MOSG: Discrete multi-objective shuffled gray wolf optimizer for multi-level image thresholding. *Engineering Science and Technology-an International Journal-Jestech, 24*(6), 1455-1466. https://doi.org/10.1016/j.jestch.2021.03.011

Karakoyun, M., Ozkis, A., & Kodaz, H. (2020). A new algorithm based on gray wolf optimizer and shuffled frog leaping algorithm to solve the multi-objective optimization problems. *Applied Soft Computing, 96*, Article 106560. https://doi.org/10.1016/j.asoc.2020.106560

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Proceedings of ICNN'95-international conference on neural networks,

Mafarja, M., & Mirjalili, S. (2018). Whale optimization approaches for wrapper feature selection. *Applied Soft Computing, 62*, 441-453.

Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, *95*, 51-67.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, *69*, 46-61.

Mirjalili, S. Z., Mirjalili, S., Saremi, S., Faris, H., & Aljarah, I. (2018). Grasshopper optimization algorithm for multi-objective optimization problems. *Applied Intelligence*, *48*(4), 805-820. https://doi.org/10.1007/s10489-017-1019-8

Mostafa, R. R., El-Attar, N. E., Sabbeh, S. F., Vidyarthi, A., & Hashim, F. A. (2022). ST-AL: a hybridized search based metaheuristic computational algorithm towards optimization of high dimensional industrial datasets. *Soft Computing*. https://doi.org/10.1007/s00500-022-07115-7

Oliva, D., & Elaziz, M. A. (2020). An improved brainstorm optimization using chaotic opposite-based learning with disruption operator for global optimization and feature selection. *Soft Computing*, *24*(18), 14051-14072. https://doi.org/10.1007/s00500-020-04781-3

Toz, M. (2019). An improved form of the ant lion optimization algorithm for image clustering problems. *Turkish Journal of Electrical Engineering and Computer Sciences*, *27*(2), 1445-1460.

Yan, S. C., Wu, L. F., Fan, J. L., Zhang, F. C., Zou, Y. F., & Wu, Y. (2021). A novel hybrid WOA-XGB model for estimating daily reference evapotranspiration using local and external meteorological data: Applications in arid and humid regions of China. *Agricultural Water Management, 244*.

Zhang, X. M., & Wen, S. C. (2021). Hybrid whale optimization algorithm with gathering strategies for high-dimensional problems. *Expert Systems with Applications, 179*, Article 115032. https://doi.org/10.1016/j.eswa.2021.115032

Zhu, G. Y., & Zhang, W. B. (2017). Optimal foraging algorithm for global optimization. *Applied Soft Computing*, *51*, 294-313.