

Received March 3, 2022, accepted May 3, 2022, date of publication May 9, 2022, date of current version May 16, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3173401

# Mud Ring Algorithm: A New Meta-Heuristic Optimization Algorithm for Solving Mathematical and Engineering Challenges

ABEER S. DESUKY<sup>1</sup>, MEHMET AKIF CIFCI<sup>2</sup>, (Member, IEEE), SAMINA KAUSAR<sup>3</sup>, SADIQ HUSSAIN<sup>4</sup>, AND LAMIAA M. EL BAKRAWY<sup>1</sup>

<sup>1</sup>Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo 11754, Egypt

<sup>2</sup>Department of Computer Engineering, Bandirma Onyedi Eylul University, 10200 Balikesir, Turkey

<sup>3</sup>Department of Computer Sciences and Information Technology, University of Kotli Azad Jammu and Kashmir, Kotli Azad Kashmir 11100, Pakistan

<sup>4</sup>Examination Branch, Dibrugarh University, Dibrugarh 786004, India

Corresponding author: Samina Kausar (saminamalik7@yahoo.com)

**ABSTRACT** This paper proposes a new meta-heuristic optimization algorithm, namely Mud Ring Algorithm (MRA) that mimics the mud ring feeding behaviour of bottlenose dolphins in the Atlantic coast of Florida. The inspiration of MRA is mainly based on the foraging behaviour of bottlenose dolphins and their mud ring feeding strategy. This strategy is applied by dolphins to trap fish via creating a plume by a single dolphin moving his tail swiftly in the sand and swims around the group of fish. The fishes become disoriented and jump over the surface only to find the waiting mouths of dolphins. MRA optimization algorithm mathematically simulates this feeding strategy and proves its optimization effectiveness through a comprehensive comparison with other meta-heuristic algorithms. Twenty-nine benchmark functions and four commonly used benchmark engineering challenges are used in the comparison. The statistical comparisons and results prove that the proposed MRA has the superiority in dealing with these optimization problems and can obtain the best solutions than other meta-heuristic optimizers.

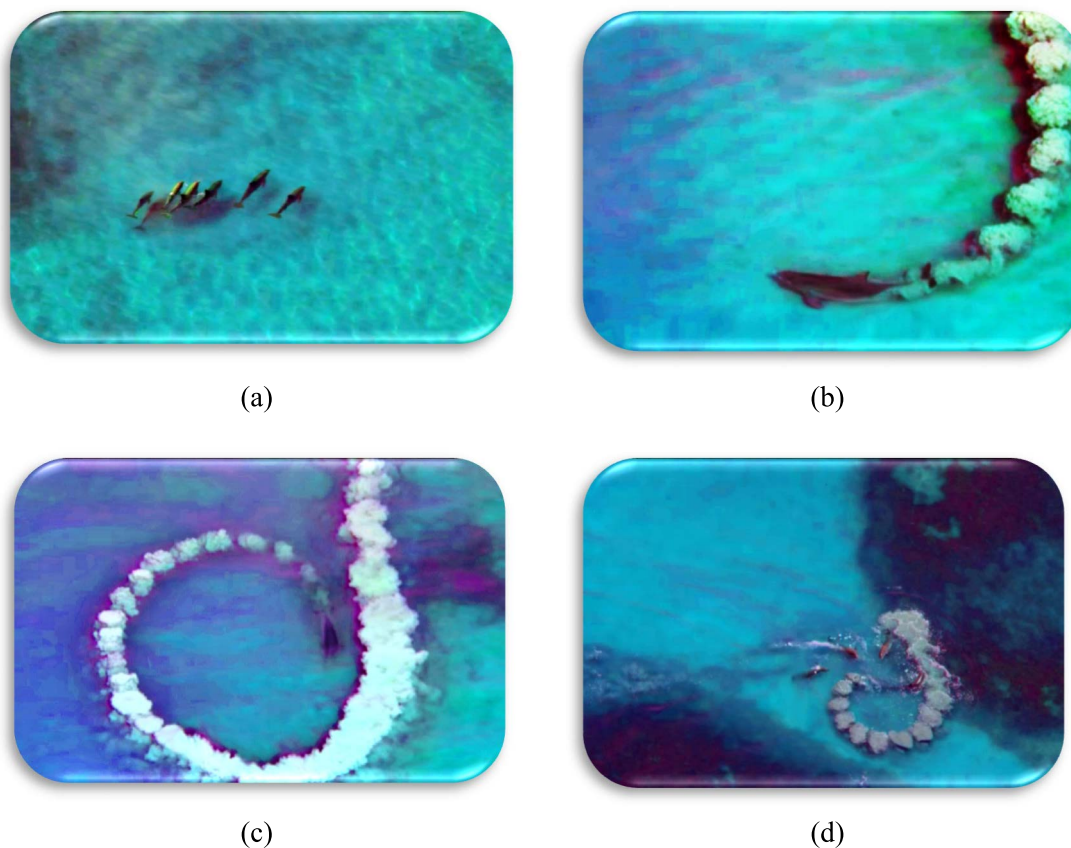
**INDEX TERMS** Optimization algorithm, meta-heuristic, nature-inspired algorithms, swarm intelligence, 3-bar truss design challenge, tension/compression spring design challenge, pressure vessel design challenge, welded beam design challenge.

## I. INTRODUCTION

Swarm intelligence has gained attention from the machine learning community since its inception [1]. Many optimization problems that are challenging to solve applying traditional techniques can be successfully tackled by swarm intelligence methods. Differences, fluctuation, negative feedback, positive feedback, self-organization, stigmergy, and decentralization are some of the main features of the swarm intelligence. Properties and definition of swarm intelligence were exploited to devise many algorithms based on it. These methods can be categorized into two types. One is bionic process algorithm, and the other is bionic behaviour algorithm. The first category is supervised by genetic algorithm and population evolution is emulated by it [2]. The second category mimics the behaviour of various species searching for prey.

The associate editor coordinating the review of this manuscript and approving it for publication was Wei-Yen Hsu<sup>1</sup>.

Ant colony optimization [3], artificial bee colony [4], particle swarm optimization [5], and dolphin swarm algorithm [1] are some of the examples of this category. These algorithms have a variety of applications in pattern recognition, cluster analysis, wireless sensor networks, and so on. Favourable performances are recorded by these algorithms and they are being widely implemented. Ant colony optimization shams the food-seeking behaviour of ants, and pheromones for information exchange are used to find the shortest path from their nest to the food source. Bees' honey gathering behaviour is being utilized by the artificial bee colony algorithm and it located better nectar sources via division of labour and cooperation. Exchanges and mutation of genes are simulated by the genetic algorithm and by the 'survival of the fittest' rule, the best environmentally suitable gene is located. Predatory behaviour of birds' swarm is mimicked by particle swarm optimization method and prey is found out by information exchange and locating the best from the search results considering the other



**FIGURE 1.** Portrayal of Mud plume feeding from overhead. a. Dolphins march together for prey b. Preliminary advent of suspended sediment c. plume propagates with the movement of dolphin d. termination of plume progression and relocation of dolphins in accordance to the plume and dolphins attacking via the plume.

birds' information. Humans attain an encouraging upshot by mimicking nature.

Apart from birds, bees, and ants, that had been simulated successfully, there are other creatures in nature such as firefly, bacteria that are worth investigating. Dolphin is no exception. Instead, the dolphin has many living habits and biological characteristics that can be simulated such as information exchanges, division of labour, cooperation, and echolocation. During their predatory process, several stages including search, call, reception, and predation come in handy to acquire their goal. In the Atlantic coast of Florida, United States, cooperative feeding behaviour of bottlenose dolphins is observed. It is called mud ring feeding. It is also dubbed mud plume fishing [6]. This hunting strategy is applied by dolphins to forage and trap fish. A plume is created by a single dolphin moving the tail swiftly in the sand and swims around a group of fish. The fishes become disoriented as it generates a momentary net around them. The fishes jump over the surface only to find dolphins that plunge through the plume to catch them. Figure 1 demonstrated “mud plume feeding” sequentially from overhead.

This work proposes a new meta-heuristic optimization technique inspired by the mud ring feeding behaviour of

bottlenose dolphins. This technique we term as ‘Mud Ring Algorithm’ (MRA).

The remaining sections of this paper are structured as follows. Section II presents a literature review that is relevant to the research of various optimization techniques addressing the engineering design challenges. Section III showcased the MRA inspiration and mathematical simulation. Section IV elaborates on the experimental results and discussion and Section V discusses the benchmark engineering challenges. Finally, we summarize the full text in Section VI.

## II. RELATED WORKS

Wu *et al.* [1] introduced a novel method named ‘dolphin swarm algorithm’ integrating living habits and biological characteristics of dolphins into swarm intelligence to tackle optimization problem. Artificial bee colony algorithm, genetic algorithm, particle swarm optimization, and the proposed dolphin swarm algorithm were employed to test ten benchmark functions. The benchmark function and convergence rates results showcased the superior performance of the dolphin swarm algorithm. It demonstrated some interesting features like no particular demand on benchmark functions, periodic, and first-slow-then-fast convergence.

The research community is always interested in finding a solution to high dimensional function. A new algorithm was proposed by Qiao *et al.* [7] to tackle high-dimensional functions more efficiently by exploiting the dolphin swarm algorithm and Kernel Fuzzy C-means. Five high-dimensional functions were utilized to examine the performance of this integrated approach. These functions achieved global optimal solutions and the integrated algorithm performed better than the dolphin swarm method and some other optimization techniques on the five applied indicators.

Many meta-heuristics algorithms are applied to solve different engineering challenges. PSO is one of such algorithms commonly used for finding near-optimal solutions and their simplicity. But it lacks the issue of balancing between local and global search abilities when used in engineering challenges. An enhanced version of PSO was proposed and it was based on a multi-swarm framework [8]. This variant was applied to Welded Beam Design (WBD) challenge. WBD is a mechanical challenge. A better optimal solution was yielded by the improved version of PSO to simulate the WBD engineering challenge. Its speedy convergence learning behaviour made it a worthy meta-heuristics algorithm.

Another optimization method termed neutrosophic optimization (NSO) was utilized to optimise the cost of welded steel beams [9]. Devising an optimal minimum cost welded beam and dimensioning the beam such as depth, width, length, and height of the beam. Based on falsity, indeterminacy, and truth membership function, a cost-effective solution was developed by the NSO method considering the flexible constraints. In imprecise and precise environment, the proposed method outperformed iterative method for non-linear WBD.

Inspired by the social behaviour of firefly, the firefly algorithm was devised. Çelik *et al.* [10] modified the firefly algorithm and acclimatized the neighbourhood technique of it to solve the Tension/Compression Spring Design challenge. Their method showcased superior results against other meta-heuristics methods and enhanced the eminence of elucidations for the engineering design challenge.

Zitouni *et al.* [11] presented a new method for global optimization dubbed the Solar System Algorithm. The objects located in the solar system such as black holes, stars, moons, and planets were imitated by the introduced method. This method was employed to solve five engineering challenges: gear train, welded beam, pressure vessel, 3-bar truss, and tension/compression spring. Their method was compared with 27 meta-heuristic algorithms and showed efficient solutions for the aforementioned five engineering challenges.

Đurđev *et al.* [12] studied different optimization techniques in the engineering domain and optimal solutions attained by these methods. Tension/compression spring design is another engineering challenge. Several popular and new swarm-based meta-heuristic methods were exploited to mitigate the weight of the spring and to achieve a competitive outcome. Statistical results, as well as representations of convergence curves, were also incorporated into their study.

YILDIRIM *et al.* [13] deployed Artificial Atom Algorithm for solving the 3-bar truss engineering challenge. This meta-heuristic algorithm called Artificial Atom Algorithm is one of chemistry-based optimization methods. They compared their outcome with Cricket Algorithm, Mine Blast Algorithm, Bat Algorithm, and Cuckoo Search Algorithm, which were utilized to solve the same challenge.

The 3-bar truss is one of the significant civil engineering challenges and a near-optimal solution had been demonstrated by meta-heuristic strategies. Fauzi *et al.* [14] examined the feasibility of tackling the 3-bar truss issue based on a novel single-solution simulated Kalman filter method. The method achieved comparable results with the hybrid lightning search algorithm-simplex technique and outperformed other methods in the literature.

Salih *et al.* [15] introduced Multi-Swarm Particle Swarm Optimization (MPSO), an enhanced version of PSO, and evaluated its efficiency for solving the engineering challenge called pressure vessel design. MPSO achieved a superior solution to the engineering challenge than the PSO with a fast convergent process.

Whale Optimization Algorithm showed low performance in the exploitation phase and deteriorated in the local optimal solution. On the contrary, Grey Wolf Optimization executed tremendous performance in the phase of exploitation while tested using the standard unimodal benchmark functions. The main motive of the study by Mohammed *et al.* [16] was to integrate these two optimization methods to handle these issues. To enhance the solution, a novel method was incorporated into the phase of exploration after each iteration. Their hybridized method was also tested to solve the pressure vessel design challenge. It outperformed the other techniques in the same domain confirmed by the Wilcoxon rank-sum test.

In this paper, a new meta-heuristic optimization technique inspired by the mud ring feeding behaviour of bottlenose dolphins is proposed. This technique we term as 'Mud Ring Algorithm' (MRA). The MRA technique addresses Twenty-nine benchmark functions and four commonly used benchmark engineering challenges- (1) 3-bar truss design challenge, (2) Tension/compression spring design challenge, (3) Pressure vessel design challenge, and (4) Welded beam design challenge. The Wilcoxon signed-rank test is also applied as statistical validation for the experimental results. The results show that the proposed optimization algorithm MRA statistically outperforms comparable meta-heuristic optimizers.

### III. MRA INSPIRATION AND MATHEMATICAL SIMULATION

#### A. INSPIRATION

Bottlenose dolphins (*Tursiops truncatus*) are one of 76 marine mammals and cetacean types. They can dive around 260 m down the surface of the ocean [17]. Common bottlenose dolphins can often be found swimming behind trawlers in various areas around the world to search for food. They generate

three types of sounds: echolocation clicks, whistles, and burst-pulse sounds. Dolphin echolocation is the generation of clicks with high frequencies (40–130) kHz. It is used in searching for food, navigation, and predator detection via the ultrasounds which bounce off objects through the water and return back to the dolphin that can detect it by its Mellon organ which acts like sonar and decodes the message. Dolphins can also generate a large form of whistles such as signature whistles which are individually specified to communicate location, identity, and probably emotional state. But through social interactions, burst-pulses sounds are generated [17], [18].

To get food, Bottlenose dolphins cooperate by working as a team to optimize the hunting effort. Many hunt strategies were used by dolphins. These strategies vary according to the prey items and environmental conditions (habitat). Mud Ring feeding (mud plume fishing) is a unique strategy of foraging which was observed for the first time during the behaviour studying of bottlenose dolphins in the shallow waters of the Atlantic coast of Florida in 1999. In this foraging method, after forming the dolphins' swarm, one dolphin from the swarm swims around the prey (fish group) to form a circle along the ocean coast floor, moving its tail up and down near the sand to create a ring or plume of mud which causing the fish to disorient and forcing them to leap out of the water to the mouths of the dolphins waiting along the outer edge of the mud ring [6].

Mud Ring Algorithm (MRA) simulates this foraging behaviour of bottlenose dolphins starting from the process of searching for the prey by the dolphins' swarm using echolocation and ending with the mud ring forming for feeding.

Before simulating the MRA algorithm mathematically, some points need an explanation as follows:

By the beginning of the hunting process, the dolphins' swarm gets closer to the prey every time, and this is shown with the  $K$  parameter. This parameter is the sound loudness that is reduced each time the swarm converges to the prey, so it controls the transition between the searching for prey (exploration) and mud ring (exploitation) processes. In the exploration process, the MRA algorithm searches globally and in the process of exploitation, it tries to investigate better solutions. If this parameter is high, the exploration process takes place in the search space, but it tends to the process of exploitation when this parameter decreases.

## B. MATHEMATICAL SIMULATION

The mathematical simulation of searching for prey and mud ring feeding is presented in this subsection. Based on the following, the fundamental phases of the Mud Ring Algorithm (MRA) may be stated as the pseudocode illustrated in Algorithm 1.

### 1) SEARCHING FOR PREY (FORAGING) - EXPLORATION PHASE

If we idealize some of the echolocation features of dolphins, we employ the following rules: All dolphins utilize

echolocation to measure distance, during searching for prey; dolphins swim randomly while using velocities  $\vec{V}$  at positions  $\vec{D}$  with a sound loudness  $\vec{K}$  to look for prey. All dolphins can automatically adjust their produced sounds loudness, based on the closeness of their prey; despite the loudness might change in numerous ways, we suppose that the loudness changes depending on the time step and the pulse rate 'r' which varies between 0 and 1, where 0 indicates no emission pulses, and 1 indicates the greatest pulse emission rate. The calculations of the vector  $\vec{K}$  are:

$$\vec{K} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (1)$$

where  $\vec{r}$  is a random vector between 0 and 1, and

$$\vec{a} = 2 \left( 1 - \frac{t}{T_{\max}} \right) \quad (2)$$

To find prey (exploration), we employ virtual dolphins (search agents) organically. In a d-dimensional parameter space, dolphins search in a random position, determined by their relative positions to one another. So, we utilize  $\vec{K}$  that varies randomly with values bigger than 1 or less than  $-1$  to drive the dolphins to diverge from each other and try to find the fittest prey. Thus, a randomly selected dolphin is chosen instead of the best dolphin. This selection mechanism and  $|\vec{K}| \geq 1$  encourage exploration and enable the MRA algorithm to undertake a global search. Following is the mathematical formulation of the MRA algorithm.

we must provide the criteria for updating the positions and velocities. The workability  $\vec{D}^t$  based on the velocity  $\vec{V}^t$  at time step  $t$  is provided by

$$\vec{D}^t = \vec{D}^{t-1} + \vec{V}^t, \quad (3)$$

where  $V$  is initialized as a random vector. Initially, each dolphin is allocated a random velocity from  $[V_{\min}, V_{\max}]$  that is selected depending on the size of the problem of interest.

### 2) MUD RING FEEDING- EXPLOITATION PHASE

After detecting the prey, dolphins can locate and surround it. The MRA method considers the target prey (optimum or near to it) as the current best solution since the location of the optimal design in the search space is not known a priori. The other dolphins will thus attempt to update their positions according to the best dolphin position once the best search agent has been determined. The following equations describe this behavior:

$$\vec{A} = \left| \vec{C}\vec{D}^{*t-1} - \vec{D}^{t-1} \right| \quad (4)$$

$$\vec{D}^t = \vec{D}^{*t-1} \cdot \sin(2\pi l) - \vec{K} \cdot \vec{A} \quad (5)$$

where  $t$  marks the current time step,  $l$  is a random number,  $\vec{C}$  and  $\vec{K}$  are coefficient vectors,  $\vec{D}$  is the dolphin position vector, and  $\vec{D}^*$  is the position vector of the best dolphin position achieved so far. Monitoring that  $\vec{D}^*$  should be adjusted in each time step if there is a better position. Note that the best dolphin moves in a circle while moving its tail swiftly in the

**Algorithm 1 :**


---

```

MRA Algorithm
Set the Initial Population of Dolphins Randomly,  $D_i$ ,  $i \in [1, 2, \dots, n]$  and Velocity  $v_i$ 
Calculate the Fitness Function of Each Dolphin
 $D^*$  = the Best Dolphin Position
While ( $t < T\_max$ )
  for  $i = 1$  to  $n$ 
    Modify  $K$ ,  $C$ ,  $a$ , and  $l$ 
    if  $|K| \geq 1$  Then
      Generate New Solutions by Modifying Velocity  $v_i$  using Eq. (3)
    Else
      /* Forming the Mud ring*/
      Update the Current Dolphin Location Using Eq. (5)
    end If
  end for
  Update the Bounds for Dolphin Outside the Search Space
  Attain the Dolphin's Fitness Functions
  Update  $D^*$  in Case of a Better Position Existence
Set  $t \rightarrow t + 1$ 
end While
Return  $D^*$  (the Best position)

```

---

sand creating the shape of the sine wave to produce a plume while the other dolphins encircle the prey.

The calculation of the vector,  $\vec{C}$  is as follows:

$$\vec{C} = 2 \cdot \vec{r} \quad (6)$$

By determining the random vector  $\vec{r}$ , any position can be reached in the search area. Thus, Eq. (5) simulates the prey encircling and helps any dolphin to justify its location close to the current best position.

The searching process of MRA starts with a population of random solutions (positions of dolphins). At each time step, dolphins justify their positions regarding either the best position located so far or a randomly selected dolphin. Thus, a parameter depends on the time step to transfer between the exploration and exploitation. When,  $|\vec{K}| < 1$  the best dolphin position is elected, while when  $|\vec{K}| \geq 1$ , a random dolphin is elected for justifying the positions of the dolphins. Note that, the MRA algorithm has only two base parameters (C and K) to be modified.

#### IV. EXPERIMENTAL RESULTS AND DISCUSSION

The MRA algorithm efficiency was tested in this section using twenty-nine benchmark functions and compared with seven recent and well-established optimization algorithms. The first 23 benchmark functions are chosen from the literature [19], [20] and the other 6 benchmark functions are utilized from the CEC 2005 special session [21]. These benchmark functions cover four main categories of benchmark landscapes: (1) unimodal functions, which possess unique global best and are able to examine the exploitation process capability and convergence rate of different optimizers; (2) multimodal functions, which have a number of

local optimum and evaluate the exploration process ability and the avoidance of local optimum; (3) fixed-dimensional multimodal functions, which have a minimum number of local optimum and a fixed number of design variables, but they provide different search spaces in comparison with multimodal functions; (4) composite functions, which are then combined, shifted, rotated, and expanded version of unimodal and multimodal functions [22], [23]. Detailed descriptions of these benchmark functions are listed in Tables 1–4. In these tables, Dim is the dimension of the function, Range represents the upper and lower limit of the search space, and  $f_{min}$  is the global least value of the function. Figure 2 demonstrates a 2D view of four benchmark functions used in this paper.

In the experiments, the results of the proposed MRA algorithm are compared with seven optimization algorithms: Honey Badger Algorithm (HBA) [24], Equilibrium Optimizer (EO) [25], Grey Wolf Optimizer (GWO) [26], Ant Lion Optimizer (ALO) [27], Whale Optimization Algorithm (WOA) [28], Harris' hawk optimization (HHO) [29], and Particle Swarm Optimization (PSO) [30]. Each algorithm was run on the benchmark functions 30 times and the average (Ave) and standard deviation (Sd) of the best optimal solution were reported. All algorithms were programmed in MATLAB R2018a using a 64-bit Windows 10 system, 2.40 GHz CPU, and 16 GB memory. The maximum number of iterations (time steps) of all algorithms and the size of the population was adjusted to be 1000 and 30, respectively. The rest parameters setting of MRA, and comparative algorithms are presented in Table 5.

In this section, to analyze the influence of the dimension sizes, the performance of the proposed MRA algorithm is evaluated using three different dimension sizes

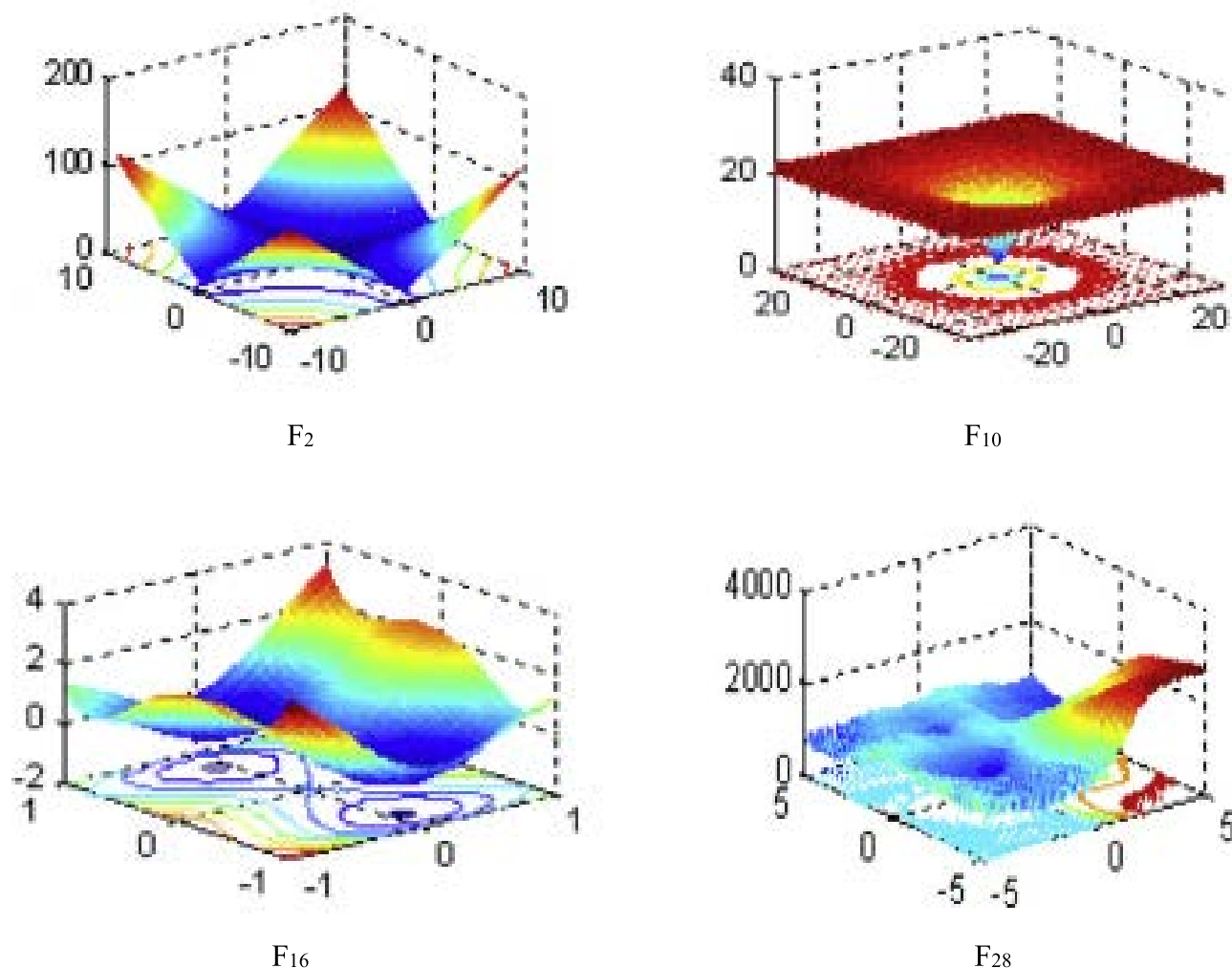


FIGURE 2. Demonstration of some selected benchmark functions.

(30, 50, and 100) of  $F_1$ – $F_{13}$  benchmark functions in addition to the  $F_{14}$ – $F_{29}$ . The experimental results for the proposed MRA algorithm against the comparative methods in dealing with  $F_1$ – $F_{13}$  benchmark functions are shown in Tables 6–8. Table 9 also shows the comparison of MRA with other algorithms in dealing with  $F_{14}$ – $F_{29}$  functions.

For 30-dimensional functions, Table 6 shows that MRA provides exact optimum values for functions  $F_1$ – $F_4$  and  $F_9$  and better results as compared to other algorithms for  $F_7$ ,  $F_{10}$  and  $F_{13}$  functions in terms of average and standard deviation. Out of thirteen benchmark functions, MRA is good for eight functions, EO and HHO are good for four functions, HBA is good for two functions, WOA for one function while GWO, ALO, and PSO fail to achieve a good result. The experimental results for 50-dimensional functions are shown in Table 7. For  $F_1$ – $F_4$  and  $F_9$  MRA obtains exact optimum values and better results for  $F_6$ ,  $F_7$ ,  $F_{10}$ ,  $F_{12}$  and  $F_{13}$  functions in terms of average and standard deviation. Table 7 also shows that MRA provides good results for ten functions out of thirteen functions, HHO for five and HBA and EO for two while

GWO, ALO, WOA, and PSO fail to obtain a good result. For 100-dimensional functions, Table 8 demonstrates that MRA can attain the exact optimum values for  $F_1$ – $F_4$  and  $F_9$  and provides the best optimal value for  $F_6$ ,  $F_7$ ,  $F_{10}$ ,  $F_{12}$  and  $F_{13}$  functions. Table 8 also shows that MRA is good for ten functions out of thirteen functions, HHO for five and HBA and EO for two while GWO, ALO, WOA, and PSO fail to obtain a good result. It can be inferred from Tables 6–8 that the proposed algorithm still provides very competitive results on  $F_1$ – $F_{13}$  benchmark functions with the increase in dimension size. It is observed from the results that the MRA is better for exploiting local area and for exploring new search space as compared to other algorithms.

As per the results shown in Table 9, MRA outperforms other algorithms and provides very competitive results on  $F_{14}$ – $F_{29}$  functions. For  $F_{16}$ ,  $F_{21}$ – $F_{23}$ ,  $F_{26}$ ,  $F_{27}$  and  $F_{29}$  MRA yields the best results compared to the other algorithms. Table 9 also demonstrates that MRA is good for seven functions, HHO for six, EO for five, ALO for three, GWO and PSO for two, and WOA for one. Convergence curves of

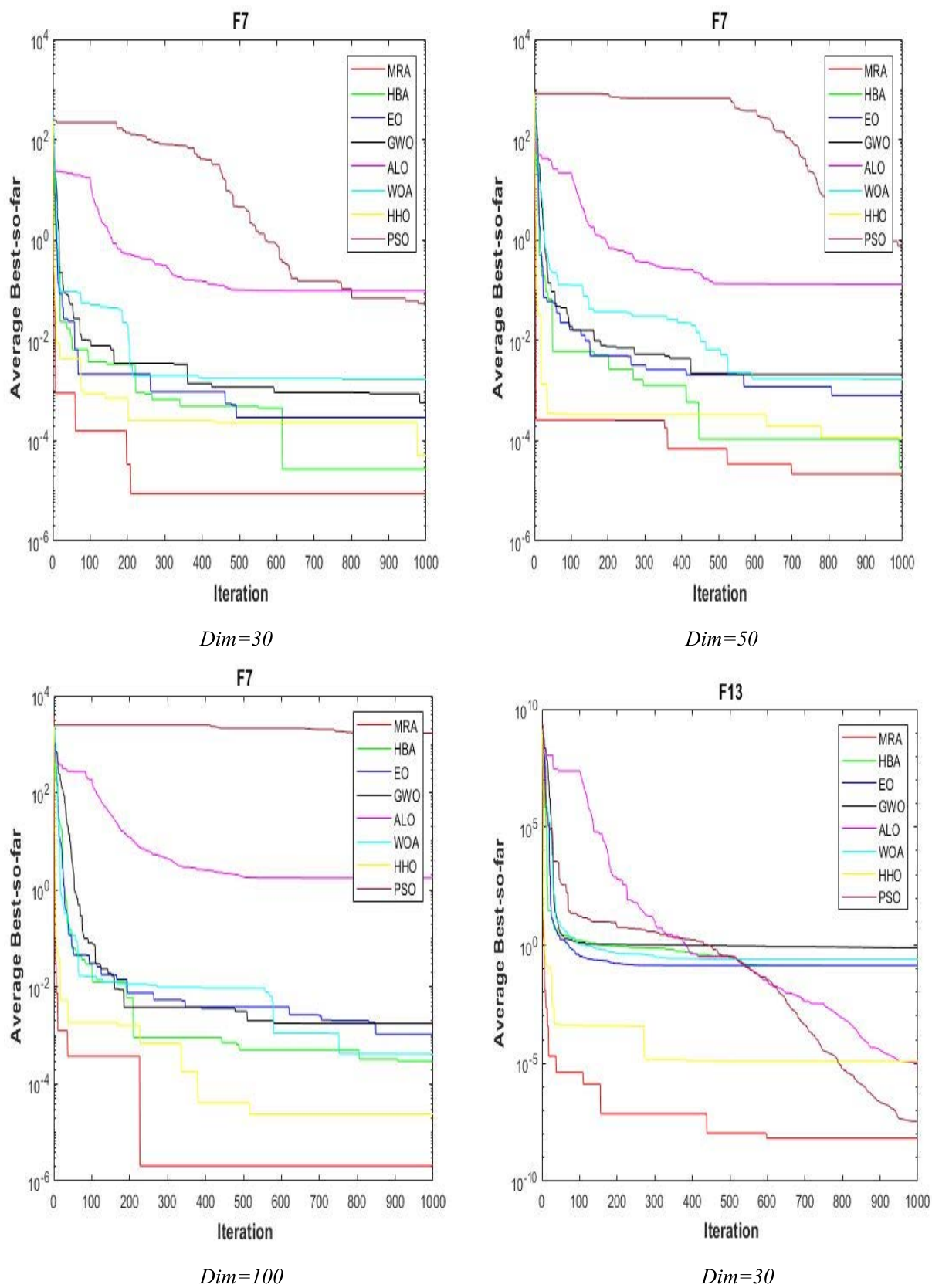


FIGURE 3. Algorithms' convergence curves for some benchmark functions.

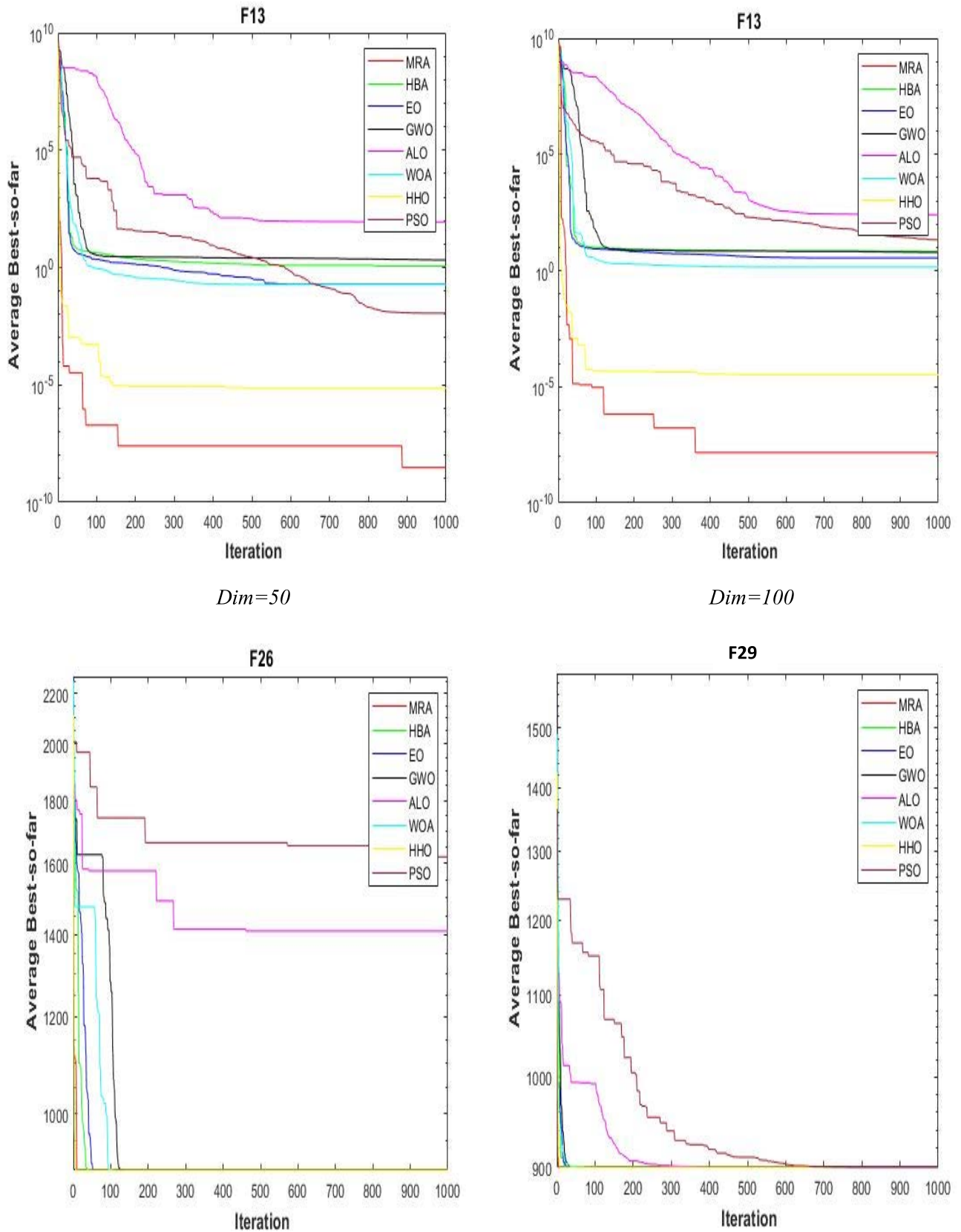


FIGURE 3. (continued.) Algorithms' convergence curves for some benchmark functions.

TABLE 1. Unimodal functions (F<sub>1</sub>-F<sub>7</sub>).

Function	Dim	Range	f <sub>min</sub>
$F_1(x) = \sum_{i=1}^n x_i^2$	30,50,100	[-100, 100]	0
$F_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30,50,100	[-10, 10]	0
$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30,50,100	[-100, 100]	0
$F_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30,50,100	[-100, 100]	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30,50,100	[-30, 30]	0
$F_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	30,50,100	[-100, 100]	0
$F_7(x) = \sum_{i=1}^n ix_i^4 + rand(0, 1)$	30,50,100	[-1.28, 1.28]	0

TABLE 2. Multimodal functions (F<sub>8</sub>-F<sub>13</sub>).

Function	Dim	Range	f <sub>min</sub>
$F_8(x) = \sum_{i=1}^n -x_i \sin \sin (\sqrt{ x_i })$	30,50,100	[-500, 500]	-418.9829×5
$F_9(x) = \sum_{i=1}^n [x_i^2 - \cos \cos (2\pi x_i) + 10]$	30,50,100	[-5.12, 5.12]	0
$F_{10}(x) = -20 \exp \exp \left( -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \exp \left( \frac{1}{n} \sum_{i=1}^n \cos \cos (2\pi x_i) \right) + 20 + e$	30,50,100	[-32, 32]	0
$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$	30,50,100	[-600, 600]	0
$F_{12}(x) = \frac{\pi}{n} \{10 \sin \sin (\prod y_i) + \sum_{i=1}^n (y_i - 1)^2 [1 + 10(\pi y_{i+1}) + (y_n - 1)^2] + \sum_{i=1}^n u(x_i, 10, 100, 4)\}$ $y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & 0 - a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30,50,100	[-50, 50]	0
$F_{13}(x) = 0.1 \{ (3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + (3\pi x_i + 1) + (x_n - 1)^2 [1 + (2\pi x_n) ] ] + \sum_{i=1}^n u(x_i, 5, 100, 4) \}$	30,50,100	[-50, 50]	0

TABLE 3. Fixed-dimension multimodal functions (F<sub>14</sub>–F<sub>23</sub>).

Function	Dim	Range	f <sub>min</sub>
$F_{14}(x) = \left( \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	2	[-65, 65]	1
$F_{15}(x) = \sum_{i=1}^{11} \left[ a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5, 5]	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	-1.0316
$F_{17}(x) = \left( x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{2}{\pi}x_1 - 6 \right)^2 + 10\left(1 - \frac{1}{8\pi}\right) \cos \cos x_1 + 10$	2	[-5, 5]	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2, 2]	3
$F_{19}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2)$	3	[1, 3]	-3.86
$F_{20}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2)$	6	[0, 1]	-3.32
$F_{21}(x) = -\sum_{i=1}^5 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.1532
$F_{22}(x) = -\sum_{i=1}^7 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.4028
$F_{23}(x) = -\sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.5363

the algorithms are plotted to test the time used by these algorithms to reach the optimum value. Figure 3 shows examples of convergence curves for some benchmark functions with 30, 50, and 100 dimensions. These curves again convey that the MRA algorithm is competitive with the increase in dimension size, and it has a faster convergence rate towards the optimum value than other algorithms.

The experimental results were also validated statistically by using the nonparametric Wilcoxon Rank-sum test [31]. A statistical comparison between the MRA optimizer and the other optimization algorithms is applied. The obtained pairwise p-values of the statistical test are presented in

Tables 10-13. In these tables, ‘NaN’ values represent ‘Not a Number’ obtained by the Wilcoxon Rank-sum test, and the p-values were compared against a 5% significance level.

Respectively, the symbols  $\vee$ ,  $\wedge$ , and  $\approx$  demonstrate that the MRA optimizer is statistically inferior, better, or similar to the state-of-the-art optimization methods that are used in the comparison. Table 10 shows the p-values for the obtained results in Table 6 and proves that the MRA algorithm is significantly better than other methods in almost all functions. Tables 11 and 12 indicate respectively the p-values for the obtained results in Tables 7 and 8. These p-values show that the solutions of the MRA algorithm are significantly superior

TABLE 4. Composite functions ( $F_{24}$ – $F_{29}$ ).

Function	Dim	Range	$f_{min}$
$F_{24}$ (CF1) $f1, f2, f3, \dots, f10$ = Sphere Function $[61, 62, 63, \dots, 610] = [1, 1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [5/100, 5/100, 5/100, \dots, 5/100]$	30	[-5,5]	0
$F_{25}$ (CF2) $f1, f2, f3, \dots, f10$ = Griewank's Function $[61, 62, 63, \dots, 610] = [1, 1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [5/100, 5/100, 5/100, \dots, 5/100]$	30	[-5,5]	0
$F_{26}$ (CF3) $f1, f2, f3, \dots, f10$ = Griewank's Function $[61, 62, 63, \dots, 610] = [1, 1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [1, 1, 1, \dots, 1]$	30	[-5,5]	0
$F_{27}$ (CF4) $f1, f2$ = Ackley's Function, $f3, f4$ = Rastrigin's Function, $f5, f6$ = Weierstrass Function $f7, f8$ = Griewank's Function, $f9, f10$ = Sphere Function $[61, 62, 63, \dots, 610] = [1, 1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [5/32, 5/32, 1, 1, 5/0.5, 5/0.5, 5/100, 5/100, 5/100, 5/100]$	30	[-5,5]	0
$F_{28}$ (CF5) $f1, f2$ = Ackley's Function, $f3, f4$ = Rastrigin's Function, $f5, f6$ = Weierstrass Function $f7, f8$ = Griewank's Function, $f9, f10$ = Sphere Function $[61, 62, 63, \dots, 610] = [1, 1, 1, \dots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$	30	[-5,5]	0
$F_{29}$ (CF6) $f1, f2$ = Ackley's Function, $f3, f4$ = Rastrigin's Function, $f5, f6$ = Weierstrass Function $f7, f8$ = Griewank's Function, $f9, f10$ = Sphere Function $[61, 62, 63, \dots, 610] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [0.1*1/5, 0.2*1/5, 0.3*5/0.5, 0.4*5/0.5, 0.5*5/100, 0.6*5/100, 0.7*5/32, 0.8*5/32, 0.9*5/100, 1*5/100]$	30	[-5,5]	0

TABLE 5. The parameter settings for MRA and comparative algorithms.

Algorithms	Parameter Settings
PSO	minimum inertia weight = 0.2 and Maximum inertia weight = 0.9 individual-best acceleration factor ( $C_1$ ) = global-best acceleration factor ( $C_2$ ) = 2
HHO	E0 variable varies from -1 to 1
WOA	$b = 1$ and $\vec{a}$ decreases linearly from 2 to 0 $\vec{a}_2$ decreases linearly from -1 to -2
ALO	w has a value from 2 to 6 depends on the current iteration
GWO	$\vec{a}$ decreases linearly from 2 to 0
EO	$\alpha_1 = 2$ and $\alpha_2 = 1$ Generation probability (GP) = 0.5
HBA	$C = 2$ and beta = 6
MRA	$\vec{r}$ in the range [0, 1]

**TABLE 6.** Experimental results for  $F_1$ – $F_{13}$  benchmark functions with 30 dimensions.

	MRA		HBA		EO		GWO		ALO		WOA		HHO		PSO	
	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd
F1	0	0	$3.13 \times 10^{-275}$	0	$1.84 \times 10^{-85}$	$9.08 \times 10^{-85}$	$9.15 \times 10^{-58}$	$2.33 \times 10^{-57}$	$1.25 \times 10^{-05}$	$9.05 \times 10^{-06}$	$5.77 \times 10^{-149}$	$3.16 \times 10^{-148}$	$2.24 \times 10^{-186}$	0	$5.27 \times 10^{-09}$	$9.50 \times 10^{-09}$
F2	0	0	$8.43 \times 10^{-146}$	$2.77 \times 10^{-145}$	$2.61 \times 10^{-49}$	$3.40 \times 10^{-49}$	$8.43 \times 10^{-35}$	$8.49 \times 10^{-35}$	$4.45 \times 10^0$	$5.04 \times 10^0$	$1.25 \times 10^{-101}$	$6.87 \times 10^{-101}$	$8.88 \times 10^{-96}$	$4.32 \times 10^{-95}$	$3.59 \times 10^{-04}$	$1.08 \times 10^{-03}$
F3	0	0	$7.31 \times 10^{-202}$	0	$1.73 \times 10^{-21}$	$7.03 \times 10^{-21}$	$2.12 \times 10^{-14}$	$1.13 \times 10^{-13}$	$1.68 \times 10^0$	$9.39 \times 10^0$	$3.36 \times 10^{04}$	$1.46 \times 10^{04}$	$1.13 \times 10^{-138}$	$6.19 \times 10^{-138}$	$1.48 \times 10^0$	$6.98 \times 10^0$
F4	0	0	$3.26 \times 10^{-112}$	$1.76 \times 10^{-111}$	$3.25 \times 10^{-20}$	$1.56 \times 10^{-19}$	$2.51 \times 10^{-14}$	$3.37 \times 10^{-14}$	$1.53 \times 10^0$	$4.99 \times 10^0$	$4.87 \times 10^{01}$	$3.20 \times 10^{01}$	$2.63 \times 10^{-92}$	$9.81 \times 10^{-92}$	$6.24 \times 10^{-01}$	$1.80 \times 10^{-01}$
F5	$2.20 \times 10^0$	$1.23 \times 10^0$	$2.18 \times 10^{01}$	$6.84 \times 10^{-01}$	$2.45 \times 10^{01}$	$1.58 \times 10^{-01}$	$2.69 \times 10^0$	$5.93 \times 10^{-01}$	$1.96 \times 10^0$	$3.61 \times 10^0$	$2.73 \times 10^{01}$	$5.49 \times 10^{-01}$	$1.22 \times 10^{-03}$	$1.33 \times 10^{-03}$	$6.42 \times 10^{-01}$	$5.68 \times 10^{-01}$
F6	$1.14 \times 10^{-08}$	$2.73 \times 10^{-08}$	$1.21 \times 10^{-06}$	$5.97 \times 10^{-06}$	$4.81 \times 10^{-09}$	$8.70 \times 10^{-09}$	$5.91 \times 10^{-01}$	$2.65 \times 10^{-01}$	$8.24 \times 10^{-06}$	$6.60 \times 10^{-06}$	$4.31 \times 10^{-02}$	$4.15 \times 10^{-02}$	$4.94 \times 10^{-05}$	$7.04 \times 10^{-05}$	$6.40 \times 10^{-07}$	$3.27 \times 10^{-06}$
F7	$4.53 \times 10^{-05}$	$4.46 \times 10^{-05}$	$1.93 \times 10^{-04}$	$1.50 \times 10^{-04}$	$5.41 \times 10^{-04}$	$3.12 \times 10^{-04}$	$6.94 \times 10^{-04}$	$3.73 \times 10^{-04}$	$1.33 \times 10^{-01}$	$6.00 \times 10^{-02}$	$2.33 \times 10^{-03}$	$2.39 \times 10^{-03}$	$5.72 \times 10^{-05}$	$4.69 \times 10^{-05}$	$7.88 \times 10^{-02}$	$2.55 \times 10^{-02}$
F8	$6.16 \times 10^0$	$1.98 \times 10^0$	$8.80 \times 10^{03}$	$1.26 \times 10^{03}$	$-8.99 \times 10^{03}$	$6.82 \times 10^0$	$6.50 \times 10^0$	$8.20 \times 10^0$	$5.54 \times 10^0$	$2.13 \times 10^0$	$1.08 \times 10^{04}$	$1.79 \times 10^{03}$	$1.25 \times 10^{04}$	$1.41 \times 10^{02}$	$6.99 \times 10^0$	$1.72 \times 10^0$
F9	0	0	0	0	0	$6.15 \times 10^{-01}$	$1.76 \times 10^0$	$8.74 \times 10^0$	$2.22 \times 10^0$	0	0	0	0	$4.54 \times 10^0$	$1.16 \times 10^0$	
F10	$8.88 \times 10^{-16}$	0	$1.99 \times 10^{00}$	$6.06 \times 10^{00}$	$5.03 \times 10^{-15}$	$1.35 \times 10^{-15}$	$1.57 \times 10^{-14}$	$3.24 \times 10^{-15}$	$2.30 \times 10^0$	$1.24 \times 10^0$	$4.09 \times 10^{-15}$	$2.53 \times 10^{-15}$	$8.88 \times 10^{-16}$	0	$3.12 \times 10^{-02}$	$1.70 \times 10^{-01}$
F11	$6.28 \times 10^0$	$2.04 \times 10^0$	0	0	0	0	$2.11 \times 10^{-03}$	$5.34 \times 10^{-03}$	$8.90 \times 10^{-03}$	$1.29 \times 10^{-02}$	$3.70 \times 10^{-18}$	$2.03 \times 10^{-17}$	0	$8.62 \times 10^{-03}$	$8.86 \times 10^{-03}$	
F12	$4.58 \times 10^{-10}$	$5.25 \times 10^{-10}$	$1.61 \times 10^{-08}$	$2.30 \times 10^{-08}$	$8.20 \times 10^{-11}$	$9.88 \times 10^{-11}$	$4.39 \times 10^{-02}$	$2.51 \times 10^{-02}$	$1.26 \times 10^0$	$5.16 \times 10^0$	$7.13 \times 10^{-03}$	$5.43 \times 10^{-03}$	$3.81 \times 10^{-06}$	$7.63 \times 10^{-06}$	$3.46 \times 10^{-03}$	$1.89 \times 10^{-02}$
F13	$8.86 \times 10^{-09}$	$1.33 \times 10^{-08}$	$1.34 \times 10^{-01}$	$1.56 \times 10^{-01}$	$2.69 \times 10^{-02}$	$4.38 \times 10^{-02}$	$4.93 \times 10^{-01}$	$1.91 \times 10^{-01}$	$5.90 \times 10^0$	$1.17 \times 10^0$	$2.00 \times 10^{-01}$	$1.51 \times 10^{-01}$	$1.41 \times 10^{-05}$	$1.77 \times 10^{-05}$	$2.56 \times 10^{-03}$	$4.73 \times 10^{-03}$

**TABLE 7.** Experimental results for  $F_1$ – $F_{13}$  benchmark functions with 50 dimensions.

	MRA		HBA		EO		GWO		ALO		WOA		HHO		PSO	
	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd
F1	0	0	$2.74 \times 10^{-264}$	0	$3.52 \times 10^{-73}$	$1.10 \times 10^{-72}$	$2.34 \times 10^{-43}$	$4.40 \times 10^{-43}$	$8.79 \times 10^{-04}$	$5.31 \times 10^{-04}$	$2.85 \times 10^{-150}$	$1.29 \times 10^{-149}$	$5.27 \times 10^{-184}$	0	$7.05 \times 10^{-04}$	$8.59 \times 10^{-04}$
F2	0	0	$2.15 \times 10^{-138}$	$8.29 \times 10^{-138}$	$2.75 \times 10^{-42}$	$5.36 \times 10^{-42}$	$4.77 \times 10^{-26}$	$4.08 \times 10^{-26}$	$1.49 \times 10^0$	$8.87 \times 10^0$	$2.67 \times 10^{-103}$	$1.22 \times 10^{-102}$	$1.21 \times 10^{-96}$	$4.03 \times 10^{-96}$	$1.66 \times 10^{-01}$	$1.55 \times 10^{-01}$
F3	0	0	$4.02 \times 10^{-184}$	0	$2.86 \times 10^{-12}$	$7.12 \times 10^{-12}$	$7.96 \times 10^{-06}$	$3.88 \times 10^{-05}$	$1.24 \times 10^0$	$3.04 \times 10^0$	$2.03 \times 10^0$	$5.33 \times 10^0$	$2.61 \times 10^{-126}$	$1.43 \times 10^{-125}$	$6.23 \times 10^0$	$1.21 \times 10^0$
F4	0	0	$1.10 \times 10^{-93}$	$3.95 \times 10^{-93}$	$6.08 \times 10^{-14}$	$1.12 \times 10^{-13}$	$6.90 \times 10^{-05}$	$3.75 \times 10^{-04}$	$2.59 \times 10^0$	$4.81 \times 10^0$	$5.97 \times 10^0$	$2.91 \times 10^0$	$4.68 \times 10^{-93}$	$1.67 \times 10^{-92}$	$2.27 \times 10^0$	$2.96 \times 10^0$
F5	$3.40 \times 10^0$	$2.26 \times 10^0$	$4.32 \times 10^0$	$1.73 \times 10^0$	$4.46 \times 10^0$	$2.32 \times 10^0$	$4.73 \times 10^0$	$8.77 \times 10^0$	$3.46 \times 10^0$	$4.52 \times 10^0$	$4.75 \times 10^0$	$5.00 \times 10^0$	$7.35 \times 10^0$	$1.22 \times 10^0$	$1.61 \times 10^0$	$8.13 \times 10^0$
F6	$2.96 \times 10^{-08}$	$4.23 \times 10^{-08}$	$1.02 \times 10^{-01}$	$1.82 \times 10^{-01}$	$1.44 \times 10^{-05}$	$1.21 \times 10^{-05}$	$2.41 \times 10^0$	$5.36 \times 10^0$	$9.31 \times 10^0$	$3.13 \times 10^0$	$5.34 \times 10^0$	$2.61 \times 10^0$	$5.76 \times 10^0$	$7.36 \times 10^0$	$2.66 \times 10^0$	$6.02 \times 10^0$
F7	$4.60 \times 10^{-05}$	$4.22 \times 10^{-05}$	$2.53 \times 10^{-04}$	$2.23 \times 10^{-04}$	$8.38 \times 10^{-04}$	$4.68 \times 10^{-04}$	$1.61 \times 10^{-03}$	$9.11 \times 10^{-04}$	$3.36 \times 10^{-01}$	$7.91 \times 10^{-02}$	$1.70 \times 10^{-03}$	$2.04 \times 10^{-03}$	$8.66 \times 10^{-05}$	$1.25 \times 10^{-04}$	$4.77 \times 10^{-01}$	$1.91 \times 10^{-01}$
F8	$1.08 \times 10^0$	$3.23 \times 10^0$	$1.43 \times 10^0$	$1.63 \times 10^0$	$1.46 \times 10^0$	$1.04 \times 10^0$	$9.67 \times 10^0$	$1.65 \times 10^0$	$9.28 \times 10^0$	$1.15 \times 10^0$	$1.81 \times 10^0$	$2.84 \times 10^0$	$2.09 \times 10^0$	$3.09 \times 10^0$	$1.03 \times 10^0$	$3.73 \times 10^0$
F9	0	0	0	0	0	0	$1.23 \times 10^0$	$2.33 \times 10^0$	$1.65 \times 10^0$	$2.59 \times 10^0$	$3.79 \times 10^0$	$2.08 \times 10^0$	0	0	$1.35 \times 10^0$	$2.69 \times 10^0$
F10	$8.88 \times 10^{-16}$	0	$2.66 \times 10^0$	$6.90 \times 10^0$	$6.57 \times 10^{-15}$	$1.77 \times 10^{-15}$	$3.37 \times 10^{-14}$	$4.14 \times 10^{-15}$	$4.93 \times 10^0$	$3.21 \times 10^0$	$4.20 \times 10^{-15}$	$2.63 \times 10^{-15}$	$8.88 \times 10^{-16}$	0	$3.56 \times 10^{-01}$	$5.83 \times 10^{-01}$
F11	$1.66 \times 10^0$	$4.46 \times 10^0$	0	0	0	0	$2.25 \times 10^0$	$5.40 \times 10^0$	$6.42 \times 10^0$	$2.09 \times 10^0$	$3.81 \times 10^0$	$2.08 \times 10^0$	0	$5.85 \times 10^0$	$5.62 \times 10^0$	
F12	$6.68 \times 10^{-10}$	$1.28 \times 10^{-09}$	$3.26 \times 10^{-03}$	$7.04 \times 10^{-03}$	$2.07 \times 10^{-03}$	$1.14 \times 10^{-02}$	$9.38 \times 10^{-02}$	$3.68 \times 10^{-02}$	$1.68 \times 10^0$	$7.05 \times 10^0$	$8.90 \times 10^0$	$4.11 \times 10^0$	$2.66 \times 10^0$	$5.64 \times 10^0$	$2.08 \times 10^0$	$4.44 \times 10^0$
F13	$1.37 \times 10^{-08}$	$2.75 \times 10^{-08}$	$1.42 \times 10^{-00}$	$4.66 \times 10^{-01}$	$8.76 \times 10^{-02}$	$1.14 \times 10^{-01}$	$1.94 \times 10^0$	$3.16 \times 10^0$	$7.00 \times 10^0$	$2.28 \times 10^0$	$6.15 \times 10^0$	$3.31 \times 10^0$	$1.94 \times 10^0$	$1.97 \times 10^0$	$8.54 \times 10^0$	$1.10 \times 10^0$

to those obtained by other algorithms for most all cases. The p-values for the attained results in Table 9 are listed in Table 13 and remarkably prove that MRA can significantly outperform the other competitors.

**V. BENCHMARK ENGINEERING CHALLENGES**

The MRA algorithm in this section is applied to four commonly used benchmark engineering challenges. The attained results of MRA are compared to several standards and modified optimizers proposed by other researchers in prior works.

**A. 3-BAR TRUSS DESIGN CHALLENGE**

This challenge might be considered one of the most investigated challenges in earlier publications [32]. The

mathematical representation of this challenge is described as the following:

For  $\vec{Y} = [y_1 y_2] = [A_1 A_2]$ ,  $1 = 100 \text{ cm}$ ,  $P = 2 \text{ KN/cm}^2$ , and  $\sigma = 2 \text{ KN/cm}^2$

$$\text{Maximize } f(\vec{Y}) = (2\sqrt{2}Y_1 + Y_2) \times 1,$$

Regarding

$$g_1(\vec{Y}) = \frac{\sqrt{2}y_1 + y_2}{\sqrt{2}y_1^2 + 2y_1y_2} P - \sigma \leq 0$$

$$g_2(\vec{Y}) = \frac{y_2}{\sqrt{2}y_1^2 + 2y_1y_2} P - \sigma \leq 0$$

$$\text{and } g_3(\vec{Y}) = \frac{1}{\sqrt{2}y_2 + y_1} P - \sigma \leq 0$$

With variable range  $0 \leq y_1, y_2 \leq 1$

TABLE 8. Experimental results for  $F_1$ – $F_{13}$  benchmark functions with 100 dimensions.

	MRA		HBA		EO		GWO		ALO		WOA		HHO		PSO	
	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd
F1	0	0	1.25×10 <sup>-249</sup>	0	4.38×10 <sup>-63</sup>	1.41×10 <sup>-62</sup>	2.98×10 <sup>-29</sup>	4.67×10 <sup>-29</sup>	1.77×10 <sup>01</sup>	9.73×10 <sup>00</sup>	4.06×10 <sup>-146</sup>	2.22×10 <sup>-145</sup>	7.77×10 <sup>-187</sup>	0	2.46×10 <sup>00</sup>	1.66×10 <sup>00</sup>
F2	0	0	6.04×10 <sup>-132</sup>	3.13×10 <sup>-131</sup>	9.05×10 <sup>-37</sup>	1.21×10 <sup>-36</sup>	6.53×10 <sup>-18</sup>	3.57×10 <sup>-18</sup>	2.34×10 <sup>06</sup>	1.28×10 <sup>07</sup>	7.43×10 <sup>-101</sup>	4.05×10 <sup>-100</sup>	1.14×10 <sup>-92</sup>	6.22×10 <sup>-92</sup>	1.10×10 <sup>01</sup>	4.07×10 <sup>00</sup>
F3	0	0	8.01×10 <sup>-168</sup>	0	4.01×10 <sup>-01</sup>	2.19×10 <sup>00</sup>	6.73×10 <sup>00</sup>	1.22×10 <sup>01</sup>	6.76×10 <sup>04</sup>	1.49×10 <sup>04</sup>	1.37×10 <sup>06</sup>	2.44×10 <sup>05</sup>	5.53×10 <sup>-118</sup>	2.97×10 <sup>-117</sup>	1.31×10 <sup>04</sup>	2.86×10 <sup>03</sup>
F4	0	0	7.46×10 <sup>-70</sup>	2.58×10 <sup>-69</sup>	5.24×10 <sup>00</sup>	1.86×10 <sup>01</sup>	1.04×10 <sup>00</sup>	1.98×10 <sup>00</sup>	3.81×10 <sup>01</sup>	5.46×10 <sup>00</sup>	7.77×10 <sup>01</sup>	1.88×10 <sup>01</sup>	2.03×10 <sup>-93</sup>	1.10×10 <sup>-92</sup>	8.21×10 <sup>00</sup>	9.58×10 <sup>-01</sup>
F5	5.55×10 <sup>01</sup>	4.94×10 <sup>01</sup>	9.51×10 <sup>01</sup>	1.69×10 <sup>00</sup>	9.50×10 <sup>01</sup>	9.19×10 <sup>01</sup>	9.76×10 <sup>01</sup>	7.65×10 <sup>-01</sup>	6.88×10 <sup>03</sup>	5.57×10 <sup>03</sup>	9.77×10 <sup>01</sup>	4.04×10 <sup>-01</sup>	<b>9.49×10<sup>-03</sup></b>	<b>1.52×10<sup>-02</sup></b>	2.24×10 <sup>03</sup>	8.36×10 <sup>02</sup>
F6	<b>3.72×10<sup>-08</sup></b>	<b>7.81×10<sup>-08</sup></b>	3.98×10 <sup>00</sup>	7.72×10 <sup>-01</sup>	1.01×10 <sup>00</sup>	3.85×10 <sup>-01</sup>	9.23×10 <sup>00</sup>	1.26×10 <sup>00</sup>	1.70×10 <sup>01</sup>	7.72×10 <sup>00</sup>	1.85×10 <sup>00</sup>	8.25×10 <sup>-01</sup>	1.52×10 <sup>-04</sup>	2.34×10 <sup>-04</sup>	2.54×10 <sup>00</sup>	1.65×10 <sup>00</sup>
F7	<b>5.07×10<sup>-05</sup></b>	<b>4.83×10<sup>-05</sup></b>	2.27×10 <sup>04</sup>	2.10×10 <sup>04</sup>	1.08×10 <sup>-03</sup>	6.80×10 <sup>-04</sup>	2.63×10 <sup>-03</sup>	1.00×10 <sup>-03</sup>	1.75×10 <sup>00</sup>	3.17×10 <sup>-01</sup>	2.08×10 <sup>-03</sup>	3.26×10 <sup>-03</sup>	6.71×10 <sup>-05</sup>	6.15×10 <sup>-05</sup>	1.92×10 <sup>03</sup>	5.13×10 <sup>02</sup>
F8	2.16×10 <sup>04</sup>	6.49×10 <sup>03</sup>	2.74×10 <sup>04</sup>	4.20×10 <sup>03</sup>	2.85×10 <sup>04</sup>	1.68×10 <sup>03</sup>	1.75×10 <sup>04</sup>	2.64×10 <sup>03</sup>	1.84×10 <sup>04</sup>	2.07×10 <sup>03</sup>	3.92×10 <sup>04</sup>	3.83×10 <sup>03</sup>	<b>4.19×10<sup>04</sup></b>	<b>8.03×10<sup>-01</sup></b>	1.80×10 <sup>04</sup>	7.43×10 <sup>03</sup>
F9	0	0	0	0	0	0	7.02×10 <sup>-01</sup>	1.70×10 <sup>00</sup>	3.51×10 <sup>02</sup>	6.65×10 <sup>01</sup>	0.00×10 <sup>00</sup>	0.00×10 <sup>00</sup>	0	0	4.87×10 <sup>02</sup>	8.37×10 <sup>01</sup>
F10	<b>8.88×10<sup>00</sup></b>	<b>8.88×10<sup>-16</sup></b>	3.32×10 <sup>00</sup>	7.54×10 <sup>00</sup>	7.99×10 <sup>-15</sup>	0.00×10 <sup>00</sup>	1.15×10 <sup>-13</sup>	9.74×10 <sup>-15</sup>	1.15×10 <sup>01</sup>	2.07×10 <sup>00</sup>	3.49×10 <sup>-15</sup>	2.63×10 <sup>-15</sup>	<b>8.88×10<sup>-16</sup></b>	0	2.70×10 <sup>00</sup>	3.21×10 <sup>-01</sup>
F11	3.70×10 <sup>00</sup>	1.91×10 <sup>01</sup>	0	0	0	0	1.57×10 <sup>-03</sup>	4.85×10 <sup>-03</sup>	1.16×10 <sup>00</sup>	1.24×10 <sup>-01</sup>	1.07×10 <sup>-02</sup>	4.09×10 <sup>-02</sup>	0	0	3.72×10 <sup>-02</sup>	2.65×10 <sup>-02</sup>
F12	<b>2.49×10<sup>-10</sup></b>	<b>4.40×10<sup>-10</sup></b>	4.39×10 <sup>02</sup>	1.63×10 <sup>-02</sup>	6.86×10 <sup>-03</sup>	3.54×10 <sup>-03</sup>	2.46×10 <sup>-01</sup>	5.23×10 <sup>-02</sup>	4.80×10 <sup>01</sup>	1.00×10 <sup>01</sup>	1.77×10 <sup>-02</sup>	8.79×10 <sup>-03</sup>	7.45×10 <sup>-07</sup>	1.20×10 <sup>06</sup>	1.58×10 <sup>00</sup>	1.03×10 <sup>00</sup>
F13	<b>1.60×10<sup>-08</sup></b>	<b>2.25×10<sup>-08</sup></b>	7.44×10 <sup>00</sup>	8.01×10 <sup>-01</sup>	4.75×10 <sup>00</sup>	1.31×10 <sup>00</sup>	6.38×10 <sup>00</sup>	4.34×10 <sup>-01</sup>	2.86×10 <sup>02</sup>	3.28×10 <sup>01</sup>	1.64×10 <sup>00</sup>	6.12×10 <sup>-01</sup>	4.69×10 <sup>-05</sup>	1.35×10 <sup>-04</sup>	1.35×10 <sup>01</sup>	8.47×10 <sup>00</sup>

TABLE 9. Experimental results for  $F_{14}$ – $F_{29}$  benchmark functions.

	MRA		HBA		EO		GWO		ALO		WOA		HHO		PSO	
	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd	Ave	Sd
F14	1.78×10 <sup>00</sup>	2.96×10 <sup>00</sup>	1.79×10 <sup>00</sup>	1.95×10 <sup>00</sup>	<b>9.98×10<sup>01</sup></b>	<b>5.83×10<sup>17</sup></b>	3.74×10 <sup>00</sup>	3.85×10 <sup>00</sup>	1.20×10 <sup>00</sup>	4.81×10 <sup>01</sup>	3.64×10 <sup>00</sup>	4.05×10 <sup>00</sup>	1.03×10 <sup>00</sup>	1.81×10 <sup>01</sup>	3.13×10 <sup>00</sup>	2.99×10 <sup>00</sup>
F15	1.02×10 <sup>03</sup>	2.99×10 <sup>04</sup>	6.25×10 <sup>03</sup>	9.53×10 <sup>03</sup>	4.47×10 <sup>03</sup>	8.09×10 <sup>03</sup>	5.69×10 <sup>03</sup>	9.00×10 <sup>03</sup>	4.10×10 <sup>03</sup>	7.40×10 <sup>03</sup>	5.79×10 <sup>04</sup>	2.98×10 <sup>04</sup>	<b>3.74×10<sup>04</sup></b>	<b>2.16×10<sup>04</sup></b>	7.97×10 <sup>04</sup>	2.43×10 <sup>04</sup>
F16	<b>1.03×10<sup>00</sup></b>	4.81×10 <sup>03</sup>	1.03×10 <sup>00</sup>	6.39×10 <sup>16</sup>	1.03×10 <sup>00</sup>	<b>6.18×10<sup>16</sup></b>	1.03×10 <sup>00</sup>	9.82×10 <sup>09</sup>	1.03×10 <sup>00</sup>	5.01×10 <sup>14</sup>	1.03×10 <sup>00</sup>	7.78×10 <sup>11</sup>	1.03×10 <sup>00</sup>	3.35×10 <sup>11</sup>	1.03×10 <sup>00</sup>	6.78×10 <sup>16</sup>
F17	4.04×10 <sup>01</sup>	5.34×10 <sup>03</sup>	<b>3.98×10<sup>01</sup></b>	0	<b>3.98×10<sup>01</sup></b>	0	3.98×10 <sup>01</sup>	9.16×10 <sup>07</sup>	3.98×10 <sup>01</sup>	4.23×10 <sup>14</sup>	3.98×10 <sup>01</sup>	1.29×10 <sup>06</sup>	3.98×10 <sup>01</sup>	1.45×10 <sup>06</sup>	<b>3.98×10<sup>01</sup></b>	0
F18	3.91×10 <sup>00</sup>	7.81×10 <sup>01</sup>	3.90×10 <sup>00</sup>	4.93×10 <sup>00</sup>	5.70×10 <sup>00</sup>	1.48×10 <sup>01</sup>	5.70×10 <sup>00</sup>	1.48×10 <sup>01</sup>	<b>3.00×10<sup>00</sup></b>	1.56×10 <sup>13</sup>	3.00×10 <sup>00</sup>	2.01×10 <sup>05</sup>	<b>3.00×10<sup>00</sup></b>	4.96×10 <sup>08</sup>	<b>3.00×10<sup>00</sup></b>	<b>7.69×10<sup>16</sup></b>
F19	3.71×10 <sup>00</sup>	7.67×10 <sup>02</sup>	3.86×10 <sup>00</sup>	2.73×10 <sup>03</sup>	3.86×10 <sup>00</sup>	<b>2.58×10<sup>15</sup></b>	3.86×10 <sup>00</sup>	3.51×10 <sup>03</sup>	3.86×10 <sup>00</sup>	1.40×10 <sup>14</sup>	3.86×10 <sup>00</sup>	3.15×10 <sup>03</sup>	<b>3.86×10<sup>00</sup></b>	2.05×10 <sup>03</sup>	3.86×10 <sup>00</sup>	2.68×10 <sup>15</sup>
F20	2.52×10 <sup>00</sup>	1.87×10 <sup>01</sup>	3.19×10 <sup>00</sup>	1.39×10 <sup>01</sup>	3.25×10 <sup>00</sup>	7.29×10 <sup>02</sup>	3.24×10 <sup>00</sup>	1.09×10 <sup>01</sup>	<b>3.26×10<sup>00</sup></b>	6.03×10 <sup>02</sup>	3.19×10 <sup>00</sup>	1.28×10 <sup>01</sup>	3.15×10 <sup>00</sup>	8.73×10 <sup>02</sup>	3.25×10 <sup>00</sup>	<b>5.83×10<sup>02</sup></b>
F21	<b>1.02×10<sup>01</sup></b>	<b>5.57×10<sup>05</sup></b>	9.34×10 <sup>00</sup>	2.49×10 <sup>00</sup>	7.61×10 <sup>00</sup>	2.59×10 <sup>00</sup>	9.13×10 <sup>00</sup>	2.07×10 <sup>00</sup>	6.86×10 <sup>00</sup>	2.83×10 <sup>00</sup>	8.79×10 <sup>00</sup>	2.29×10 <sup>00</sup>	5.05×10 <sup>00</sup>	1.38×10 <sup>03</sup>	7.54×10 <sup>00</sup>	2.92×10 <sup>00</sup>
F22	<b>1.04×10<sup>01</sup></b>	<b>3.60×10<sup>-05</sup></b>	8.75×10 <sup>00</sup>	3.06×10 <sup>00</sup>	8.09×10 <sup>00</sup>	2.96×10 <sup>00</sup>	1.04×10 <sup>01</sup>	2.03×10 <sup>04</sup>	7.61×10 <sup>00</sup>	3.12×10 <sup>00</sup>	8.98×10 <sup>00</sup>	2.55×10 <sup>00</sup>	5.42×10 <sup>00</sup>	1.28×10 <sup>00</sup>	8.71×10 <sup>00</sup>	2.66×10 <sup>00</sup>
F23	<b>1.05×10<sup>01</sup></b>	<b>1.23×10<sup>05</sup></b>	8.80×10 <sup>00</sup>	3.22×10 <sup>00</sup>	8.19×10 <sup>00</sup>	2.99×10 <sup>00</sup>	1.04×10 <sup>01</sup>	9.87×10 <sup>01</sup>	7.50×10 <sup>00</sup>	3.37×10 <sup>00</sup>	8.19×10 <sup>00</sup>	3.18×10 <sup>00</sup>	5.48×10 <sup>00</sup>	1.34×10 <sup>00</sup>	9.74×10 <sup>00</sup>	2.10×10 <sup>00</sup>
F24	8.15×10 <sup>02</sup>	6.77×10 <sup>01</sup>	8.23×10 <sup>02</sup>	<b>3.99×10<sup>01</sup></b>	8.27×10 <sup>02</sup>	7.94×10 <sup>01</sup>	7.58×10 <sup>02</sup>	9.32×10 <sup>01</sup>	<b>6.65×10<sup>02</sup></b>	7.84×10 <sup>01</sup>	6.89×10 <sup>02</sup>	1.38×10 <sup>02</sup>	7.52×10 <sup>02</sup>	1.14×10 <sup>02</sup>	7.14×10 <sup>02</sup>	5.95×10 <sup>01</sup>
F25	8.83×10 <sup>02</sup>	1.58×10 <sup>01</sup>	8.48×10 <sup>02</sup>	5.18×10 <sup>01</sup>	8.73×10 <sup>02</sup>	<b>1.13×10<sup>01</sup></b>	7.89×10 <sup>02</sup>	1.08×10 <sup>02</sup>	<b>6.73×10<sup>02</sup></b>	7.12×10 <sup>01</sup>	7.40×10 <sup>02</sup>	1.25×10 <sup>02</sup>	8.55×10 <sup>02</sup>	6.15×10 <sup>01</sup>	7.30×10 <sup>02</sup>	9.39×10 <sup>01</sup>
F26	900	0	900	0	900	0	9.82×10 <sup>02</sup>	2.03×10 <sup>02</sup>	1.42×10 <sup>03</sup>	5.56×10 <sup>01</sup>	9.95×10 <sup>02</sup>	2.01×10 <sup>02</sup>	900	0	1.47×10 <sup>03</sup>	7.09×10 <sup>01</sup>
F27	900	0	900	0	900	0	900	0	1.04×10 <sup>03</sup>	3.11×10 <sup>01</sup>	9.31×10 <sup>02</sup>	9.72×10 <sup>01</sup>	900	0	5.36×10 <sup>03</sup>	1.06×10 <sup>02</sup>
F28	8.89×10 <sup>02</sup>	<b>2.82×10<sup>01</sup></b>	8.63×10 <sup>02</sup>	8.99×10 <sup>01</sup>	8.63×10 <sup>02</sup>	1.08×10 <sup>02</sup>	8.54×10 <sup>02</sup>	6.77×10 <sup>01</sup>	<b>7.14×10<sup>02</sup></b>	1.20×10 <sup>02</sup>	7.58×10 <sup>02</sup>	1.09×10 <sup>02</sup>	8.00×10 <sup>02</sup>	1.28×10 <sup>02</sup>	8.16×10 <sup>02</sup>	8.95×10 <sup>01</sup>
F29	900	0	900	0	900	0	<b>9.00×10<sup>02</sup></b>	1.37×10 <sup>13</sup>	9.00×10 <sup>02</sup>	5.03×10 <sup>02</sup>	900	5.36×10 <sup>14</sup>	900	0	9.00×10 <sup>02</sup>	1.34×10 <sup>-02</sup>

The view of the structured truss and the forces related to it are demonstrated in Figure 4. According to the mathematical formulation and the construction in Figure 4, There are two parameters: the first and third bars' areas and the second bar's area. The objective here is to minimize the weight of the total structure. Besides, the numerous constraints of deflection, stress, and buckling are included in this design challenge.

A total of 30 runs, each with 30 dolphins and 1000 iterations, are used to test the MRA's efficacy in this challenge. The optimization results of MRA are compared to those published for HHO [29], GOA [32], MVO [33], MFO [34], SSA [35], MBA [36], CS [37], and Ray and Sain [38] in prior research. Table 14 compares the results of the proposed MRA to preceding approaches and provides a breakdown of

TABLE 10. Pairwise Wilcoxon rank-sum test p-values (30 Dim).

	HBA	EO	GWO	ALO	WOA	HHO	PSO
F1	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^
F2	$6.87 \times 10^{-07}$ ^	$6.87 \times 10^{-07}$ ^	$6.87 \times 10^{-07}$ ^	$6.87 \times 10^{-07}$ ^	$6.87 \times 10^{-07}$ ^	$6.87 \times 10^{-07}$ ^	$6.87 \times 10^{-07}$ ^
F3	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^
F4	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^	$1.21 \times 10^{-12}$ ^
F5	$9.51 \times 10^{-06}$ v	$9.51 \times 10^{-06}$ v	$9.51 \times 10^{-06}$ v	$6.57 \times 10^{-02}$ v	$2.25 \times 10^{-04}$ v	$7.22 \times 10^{-06}$ v	$6.57 \times 10^{-02}$ v
F6	$4.12 \times 10^{-06}$ ^	$2.46 \times 10^{-01}$ ≈	$3.02 \times 10^{-11}$ ^	$3.02 \times 10^{-11}$ ^	$3.02 \times 10^{-11}$ ^	$5.49 \times 10^{-11}$ ^	$3.55 \times 10^{-01}$ ≈
F7	$1.19 \times 10^{-06}$ ^	$3.02 \times 10^{-11}$ ^	$3.02 \times 10^{-11}$ ^	$3.02 \times 10^{-11}$ ^	$6.70 \times 10^{-11}$ ^	$6.79 \times 10^{-02}$ ^	$3.02 \times 10^{-11}$ ^
F8	$3.98 \times 10^{-10}$ v	$4.39 \times 10^{-10}$ v	$6.80 \times 10^{-05}$ v	$6.83 \times 10^{-03}$ v	$2.57 \times 10^{-11}$ v	$5.22 \times 10^{-12}$ v	$3.30 \times 10^{-01}$ ≈
F9	NaN ^	NaN ^	$5.54 \times 10^{-03}$ ^	$1.21 \times 10^{-12}$ ^	NaN ^	NaN ^	$1.21 \times 10^{-12}$ ^
F10	$8.15 \times 10^{-02}$ ≈	$8.64 \times 10^{-14}$ ^	$2.54 \times 10^{-13}$ ^	$1.21 \times 10^{-12}$ ^	$9.83 \times 10^{-08}$ ^	NaN ^	$1.21 \times 10^{-12}$ ^
F11	$2.16 \times 10^{-02}$ v	$2.16 \times 10^{-02}$ v	$8.32 \times 10^{-02}$ ≈	$4.25 \times 10^{-07}$ ^	$8.32 \times 10^{-02}$ ≈	$2.16 \times 10^{-02}$ v	$4.60 \times 10^{-07}$ ^
F12	$2.03 \times 10^{-09}$ ^	$1.33 \times 10^{-02}$ v	$3.02 \times 10^{-11}$ ^	$3.02 \times 10^{-11}$ ^	$3.02 \times 10^{-11}$ ^	$3.02 \times 10^{-11}$ ^	$2.39 \times 10^{-04}$ v
F13	$4.08 \times 10^{-11}$ ^	$5.26 \times 10^{-04}$ ^	$3.02 \times 10^{-11}$ ^	$3.02 \times 10^{-11}$ ^	$3.02 \times 10^{-11}$ ^	$3.02 \times 10^{-11}$ ^	$3.40 \times 10^{-01}$ ≈

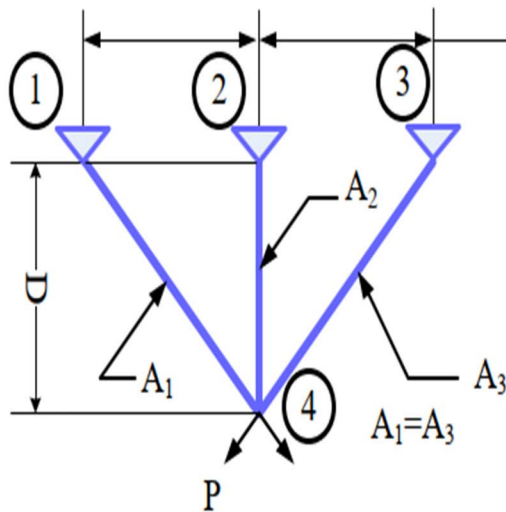


FIGURE 4. 3-bar truss design.

the findings. According to the results presented in Table 14, it is noticed that MRA may disclose extremely competitive results compared to HHO, MVO, and SSA algorithms. Other optimizers cannot compete with the MRA’s performance. The results collected prove that the MRA is capable of coping with the constrained challenge.

**B. TENSION/COMPRESSION SPRING DESIGN CHALLENGE**

In this challenge, the objective is to decrease the spring’s weight. The diameter of the mean coil (D), the diameter of the wire (d), and the active coils’ number (N) are the design variables included in this case. In addition, for this case, the constraints that are on the frequency of surge, shear stress minimum, and deflection should be met during the weight

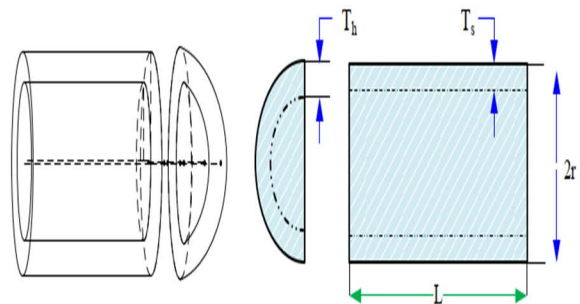


FIGURE 5. Pressure vessel design.

optimization process. Therefore, the goal and associated constraints related to this task might be expressed as:

$$\text{For } \vec{u} = [u_1 u_2 u_3] = [dDN]$$

$$\text{Minimize } f(\vec{u}) = (u_3 + 2) u_2 u_1^2$$

Regarding

$$g_1(\vec{u}) = 1 - \frac{u_2^2 u_3}{71785 u_1^4} \leq 0$$

$$g_2(\vec{u}) = \frac{4u_2^2 - u_1 u_2}{12566 (u_2 u_1^3 - u_1^4)} + \frac{1}{5108 u_1^2} \leq 0$$

$$g_3(\vec{u}) = 1 - \frac{140.45 u_1}{u_2^2 u_3} \leq 0$$

$$g_4(\vec{u}) = \frac{u_1 + u_2}{1.5} - 1 \leq 0$$

As seen in Table 15; There are numerous optimizers already used on this case like the HHO [29], MFO [34], SSA [35], TEO [39], WOA [28], methods presented by GSA [40], CPSO [41], ES [42] and RO [43]. The results of

TABLE 11. Pairwise Wilcoxon rank-sum test p-values (50 Dim).

	HBA	EO	GWO	ALO	WOA	HHO	PSO
F1	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^
F2	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^
F3	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^
F4	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^
F5	$4.73 \times 10^{-01}$ ≈	$4.73 \times 10^{-01}$ ≈	$4.73 \times 10^{-01}$ ≈	$1.83 \times 10^{-04}$ ^	$3.07 \times 10^{-01}$ ≈	$4.73 \times 10^{-01}$ ≈	$5.83 \times 10^{-04}$ ^
F6	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$5.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^
F7	$9.11 \times 10^{-03}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ v	$1.83 \times 10^{-04}$ ^	$1.71 \times 10^{-03}$ ^	$6.40 \times 10^{-02}$ ≈	$1.83 \times 10^{-04}$ ^
F8	$2.64 \times 10^{-03}$ v	$2.64 \times 10^{-03}$ v	$2.92 \times 10^{-01}$ ≈	$8.82 \times 10^{-01}$ ≈	$9.03 \times 10^{-04}$ v	$1.11 \times 10^{-04}$ v	$6.12 \times 10^{-01}$ ≈
F9	NaN ^	NaN ^	$7.79 \times 10^{-02}$ ≈	$6.39 \times 10^{-05}$ ^	NaN ^	NaN ^	$6.39 \times 10^{-05}$ ^
F10	$3.68 \times 10^{-01}$ ≈	$4.04 \times 10^{-05}$ ^	$5.35 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$5.66 \times 10^{-03}$ ^	NaN ^	$6.39 \times 10^{-05}$ ^
F11	$1.68 \times 10^{-01}$ ≈	$1.68 \times 10^{-01}$ ≈	$1.68 \times 10^{-01}$ ≈	$2.04 \times 10^{-03}$ ^	$5.84 \times 10^{-01}$ ≈	$1.68 \times 10^{-01}$ ^	$2.04 \times 10^{-03}$ ^
F12	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^
F13	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^

TABLE 12. Pairwise Wilcoxon rank-sum test p-values (100 Dim).

	HBA	EO	GWO	ALO	WOA	HHO	PSO
F1	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^
F2	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^
F3	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^
F4	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^
F5	$3.45 \times 10^{-01}$ ≈	$1.40 \times 10^{-01}$ ≈	$6.78 \times 10^{-01}$ ≈	$1.83 \times 10^{-04}$ ^	$6.78 \times 10^{-01}$ ≈	$1.40 \times 10^{-01}$ ≈	$1.83 \times 10^{-04}$ ^
F6	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^
F7	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$9.10 \times 10^{-01}$ ≈	$1.83 \times 10^{-04}$ ^
F8	$3.80 \times 10^{-04}$ v	$1.19 \times 10^{-03}$ v	$2.92 \times 10^{-01}$ ≈	$6.70 \times 10^{-01}$ ≈	$1.11 \times 10^{-04}$ v	$1.11 \times 10^{-04}$ v	$6.68 \times 10^{-01}$ ≈
F9	NaN ^	NaN ^	$6.02 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$3.68 \times 10^{-01}$ ≈	NaN ^	$6.39 \times 10^{-05}$ ^
F10	$3.50 \times 10^{-02}$ ^	$3.29 \times 10^{-05}$ ^	$4.92 \times 10^{-05}$ ^	$6.39 \times 10^{-05}$ ^	$1.56 \times 10^{-04}$ ^	NaN ^	$6.39 \times 10^{-05}$ ^
F11	$1.68 \times 10^{-01}$ ≈	$1.68 \times 10^{-01}$ ≈	$5.04 \times 10^{-01}$ ≈	$2.12 \times 10^{-02}$ ^	$1.68 \times 10^{-01}$ ≈	$1.68 \times 10^{-01}$ ≈	$2.12 \times 10^{-02}$ ^
F12	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^
F13	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^	$1.83 \times 10^{-04}$ ^

mentioned approaches are compared to the proposed MRA in Table 15.

Table 15 illustrates that the results of MRA are quite competitive with the results of HHO, TEO, and MFO. Furthermore, the proposed MRA may attain high-quality solutions in an efficient way when solving this benchmark challenge and reveals the optimal design.

C. PRESSURE VESSEL DESIGN CHALLENGE

This well-known case contains four variables and constraints, and our goal is to reduce the manufacturing cost. The used variables here are (x1 - x4): Th (x2, the head thickness),

Ts (x1, the shell thickness), r (x3, inner radius), and L (x4, the section length without the head). Figure 5 depicts the problem’s entire setup. This challenge can be formulated as:

$$\text{For } \vec{u} = [u_1 u_2 u_3 u_4] = [T_s T_h r L]$$

$$\text{Minimize } f(\vec{u}) = 0.6224u_1 u_3 u_4 + 1.7781u_2 u_3^2 + 3.1661u_1^2 u_4 + 19.84u_1^2 u_3$$

Regarding

$$g_1(\vec{u}) = -u_1 + 0.0193u_3 \leq 0$$

$$, g_2(\vec{u}) = -u_3 + 0.00954u_3 \leq 0$$

TABLE 13. Pairwise Wilcoxon rank-sum test p-values (F14–F29).

	HBA	×100	GWO	ALO	WOA	HHO	PSO
F14	1.08×10 <sup>-07</sup> v	7.84×10 <sup>-11</sup> v	5.69×10 <sup>-01</sup> ≈	5.17×10 <sup>-04</sup> ∧	3.18×10 <sup>-01</sup> ≈	2.31×10 <sup>-06</sup> ∧	6.62×10 <sup>-01</sup> ≈
F15	5.08×10 <sup>-03</sup> v	2.99×10 <sup>-04</sup> v	8.88×10 <sup>-01</sup> ≈	7.73×10 <sup>-01</sup> ≈	1.30×10 <sup>-01</sup> ≈	3.02×10 <sup>-11</sup> v	1.91×10 <sup>-01</sup> ≈
F16	5.14×10 <sup>-12</sup> v	5.14×10 <sup>-12</sup> v	3.02×10 <sup>-11</sup> v	3.02×10 <sup>-11</sup> v	3.02×10 <sup>-11</sup> v	3.02×10 <sup>-11</sup> v	3.15×10 <sup>-12</sup> v
F17	1.21×10 <sup>-12</sup> v	1.21×10 <sup>-12</sup> v	3.02×10 <sup>-11</sup> v	2.99×10 <sup>-11</sup> v	3.02×10 <sup>-11</sup> v	3.00×10 <sup>-11</sup> v	1.21×10 <sup>-12</sup> v
F18	9.05×10 <sup>-07</sup> v	1.01×10 <sup>-11</sup> v	3.02×10 <sup>-11</sup> v	3.02×10 <sup>-11</sup> v	3.02×10 <sup>-11</sup> v	3.02×10 <sup>-11</sup> v	9.36×10 <sup>-12</sup> v
F19	7.80×10 <sup>-12</sup> v	7.69×10 <sup>-12</sup> v	3.02×10 <sup>-11</sup> v	3.00×10 <sup>-11</sup> v	3.02×10 <sup>-11</sup> v	3.02×10 <sup>-11</sup> v	1.21×10 <sup>-12</sup> v
F20	8.39×10 <sup>-11</sup> v	1.69×10 <sup>-11</sup> v	3.69×10 <sup>-11</sup> v	3.02×10 <sup>-11</sup> v	4.08×10 <sup>-11</sup> v	8.99×10 <sup>-11</sup> v	1.41×10 <sup>-11</sup> v
F21	6.37×10 <sup>-07</sup> v	9.35×10 <sup>-01</sup> v	3.02×10 <sup>-11</sup> ∧	1.00×10 <sup>00</sup> ≈	3.02×10 <sup>-11</sup> ∧	3.02×10 <sup>-11</sup> ∧	6.62×10 <sup>-01</sup> ≈
F22	3.07×10 <sup>-09</sup> v	7.67×10 <sup>-02</sup> v	2.37×10 <sup>-10</sup> ∧	3.79×10 <sup>-01</sup> ≈	3.34×10 <sup>-11</sup> ∧	3.02×10 <sup>-11</sup> ∧	6.61×10 <sup>-01</sup> ≈
F23	1.22×10 <sup>-01</sup> ≈	4.69×10 <sup>-05</sup> v	1.60×10 <sup>-04</sup> ∧	3.61×10 <sup>-01</sup> ≈	4.14×10 <sup>-06</sup> ∧	3.39×10 <sup>-06</sup> ∧	1.26×10 <sup>-06</sup> v
F24	7.34×10 <sup>-01</sup> ≈	5.71×10 <sup>-01</sup> ≈	2.12×10 <sup>-01</sup> ≈	1.01×10 <sup>-03</sup> v	2.57×10 <sup>-02</sup> v	2.73×10 <sup>-01</sup> ≈	4.59×10 <sup>-03</sup> v
F25	1.97×10 <sup>-01</sup> ≈	3.44×10 <sup>-01</sup> ≈	2.17×10 <sup>-03</sup> v	1.79×10 <sup>-04</sup> v	3.36×10 <sup>-02</sup> v	6.16×10 <sup>-01</sup> ≈	1.79×10 <sup>-04</sup> v
F26	NaN ∧	NaN ∧	4.88×10 <sup>-05</sup> ∧	6.39×10 <sup>-05</sup> ∧	NaN ∧	NaN ∧	6.39×10 <sup>-05</sup> ∧
F27	NaN ∧	NaN ∧	NaN ∧	6.39×10 <sup>-05</sup> ∧	NaN ∧	NaN ∧	6.39×10 <sup>-05</sup> ∧
F28	4.51×10 <sup>-01</sup> ≈	7.01×10 <sup>-01</sup> ≈	3.51×10 <sup>-02</sup> v	3.68×10 <sup>-04</sup> v	5.08×10 <sup>-02</sup> ≈	1.28×10 <sup>-02</sup> v	2.48×10 <sup>-03</sup> v
F29	3.68×10 <sup>-01</sup> ≈	NaN ∧	9.66×10 <sup>-05</sup> ∧	6.39×10 <sup>-05</sup> ∧	1 ≈	NaN ∧	6.39×10 <sup>-05</sup> ∧

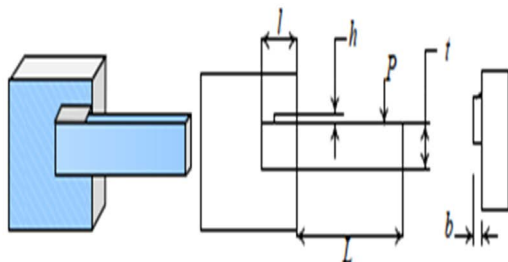


FIGURE 6. Welded beam design.

$$g_3(\vec{u}) = -\Pi u_3^2 u_4 - \frac{4}{3} \Pi u_3^3 + 1, 296, 000 \leq 0$$

$$g_4(\vec{u}) = u_4 - 240 \leq 0$$

There is just a little amount of space for design in this particular case where  $0 \leq u_2 \leq 99$  and  $0 \leq u_3, u_4 \leq 200$ . MRA's results are compared to the results of HHO [29], GA [44], GWO [26], G-QPSO [45], HPSO [46], IACO [47], WEO [48], BA [49], MFO [34], CSS [50]. Table 16 displays the best results achieved by MRA and mentioned optimizers. After examining the results shown in Table 16, we noticed that MRA has the superiority in dealing with this case and the ability to obtain the best solutions than other optimizers.

**D. WELDED BEAM DESIGN CHALLENGE**

The goal of this well-known engineering challenge is to find the minimum production cost concerning a series of constraints related to its design. A schematic depiction of this

TABLE 14. The 3-bar truss design challenge results' comparison.

Algorithm	y <sub>1</sub>	y <sub>2</sub>	Optimal cost
MRA	0.788574	0.408536	263.8959
HHO [29]	0.788663	0.408283	263.8958
GOA [32]	0.788898	0.40762	263.8959
MVO [33]	0.788603	0.408453	263.8958
MFO [34]	0.788245	0.409467	263.896
SSA [35]	0.788665	0.408276	263.8958
MBA [36]	0.788565	0.40856	263.8959
CS [37]	0.78867	0.40902	263.9716
Ray and Sain [38]	0.795	0.395	264.3

challenge is presented in Figure 6. The variables of the design include the weld length (l), thickness (h), bar thickness (b), and height (t). Therefore, the formulation of this challenge is as follows:

For  $\vec{u} = [u_1, u_2, u_3, u_4] = [h, l, t, b]$   
 Minimize  $f(\vec{u}) = 1.10471u_1^2 u_2 + 0.04811u_3 u_4 (14.0 + u_2)$

Regarding

$$g_1(\vec{u}) = \tau(\vec{u}) - \tau_{max} \leq 0$$

$$g_2(\vec{u}) = \sigma(\vec{u}) - \sigma_{max} \leq 0$$

$$g_3(\vec{u}) = \delta(\vec{u}) - \delta_{max} \leq 0$$

$$g_4(\vec{u}) = u_1 - u_4 \leq 0$$

$$g_5(\vec{u}) = P - P_c(\vec{u}) \leq 0$$

$$g_6(\vec{u}) = 0.125 - u_1 \leq 0$$

TABLE 15. The tension/compression spring results' comparison.

Algorithms	d	D	N	Optimal cost
MRA	0.052113	0.366963	10.7152	0.012672
HHO [29]	0.051796	0.359305	11.13886	0.012665
MFO [34]	0.051994	0.364109	10.86842	0.012667
SSA [35]	0.051207	0.345215	12.00403	0.012676
TEO [39]	0.051775	0.358792	11.16839	0.012665
WOA [28]	0.051207	0.345215	12.00403	0.012676
GSA [40]	0.050276	0.32368	13.52541	0.012702
CPSO [41]	0.051728	0.357644	11.24454	0.012675
ES [42]	0.051989	0.363965	10.89052	0.012681
RO [43]	0.05137	0.349096	11.76279	0.012679

TABLE 16. The pressure vessel design results' comparison.

Algorithms	T <sub>s</sub>	T <sub>h</sub>	L	r	Optimal cost
MRA	0.7812	0.3893	200	40.3197	5920.00
HHO [29]	0.817584	0.407293	176.7196	42.09175	6000.46259
GWO [26]	0.8125	0.4345	176.7587	42.08918	6051.5639
GA [44]	0.8125	0.4375	176.6541	42.0974	6059.9463
G-QPSO [45]	0.8125	0.4375	176.6372	42.0984	6059.7208
HPSO [46]	0.8125	0.4375	176.6366	42.0984	6059.7143
IACO [47]	0.8125	0.4375	176.6378	42.09835	6059.7258
WEO [48]	0.8125	0.4375	176.6366	42.09844	6059.71
BA [49]	0.8125	0.4375	176.6366	42.09845	6059.7143
MFO [34]	0.8125	0.4375	176.6366	42.09845	6059.7143
CSS [50]	0.8125	0.4375	176.5727	42.10362	6059.0888

TABLE 17. The welded beam design results' comparison.

Algorithm	l	h	b	t	Optimal cost
MRA	3.47852	0.205325	0.205714	9.039725	1.72572728
HHO [29]	3.531061	0.204039	0.206147	9.027463	1.73199057
HS [51]	6.2231	0.2442	0.2443	8.2915	2.3807
CDE [52]	3.542998	0.203137	0.206179	9.033498	1.733462
WOA [28]	3.484293	0.205396	0.206276	9.037426	1.730499
GSA [40]	3.856979	0.182129	0.202376	10	1.879952
David [53]	6.2552	0.2434	0.2444	8.2915	2.3841
Random [53]	4.7313	0.4575	0.66	5.0853	4.1185
GA2 [54]	6.173	0.2489	0.2533	8.1789	2.4331

,  $g_7(\vec{u}) = 1.10471u_1^2 + 0.04811u_3u_4(14.0 + u_2) - 5.0 \leq 0$  where,

Range of variables

$0.05 \leq u_1 \leq 2.00, 0.25 \leq u_2 \leq 1.30, 2.00 \leq u_3 \leq 15.0$

$$\tau(\vec{u}) = \sqrt{\tau'^2 + 2\tau'\tau''\frac{u_2}{2R} + \tau''^2}, \tau' = \frac{P}{\sqrt{2}u_1u_2},$$

$$\tau'' = \frac{MR}{J}, M = P \left( L + \frac{u_2}{2} \right)$$

$$, R = \sqrt{\frac{u_2^2}{4} + \left( \frac{u_1 + u_3}{2} \right)^2},$$

$$J = 2 \left\{ \sqrt{2}u_1u_2 \left[ \frac{u_2^2}{12} + \left( \frac{u_1 + u_3}{2} \right)^2 \right] \right\}, \sigma(\vec{u}) = \frac{6PL}{u_4u_2^3}$$

$$, \delta(\vec{u}) = \frac{4PL^3}{Eu_3^3u_4}, P_c(\vec{u}) = \frac{4.013E\sqrt{\frac{u_3^2u_4^6}{36}}}{L^2} \left( 1 - \frac{u_3}{2L} \sqrt{\frac{E}{4G}} \right)$$

$$, P = 6000\text{lb}, L = 14\text{in}, E = 30 \times 10^6\text{psi}, \text{ and}$$

$$G = 12 \times 10^6\text{psi}$$

In Table 17, the obtained results of MRA in comparison to the results of HHO [29], HS [51], CDE [52], and other algorithms like WOA [28], GSA [40], David [53], Random [53], and GA2 [54] show that the proposed MRA can achieve the ideal design variables with the optimal cost.

## VI. CONCLUSION

This work proposes a new meta-heuristic optimization algorithm, namely Mud Ring Algorithm (MRA). The MRA proposed Algorithm mimics the foraging behaviour (searching for prey) and the mud ring feeding strategy of bottlenose dolphins. The performance of MRA was evaluated using twenty-nine benchmark functions. The MRA ability to exploit the local area, explore new search space, and avoid local optima was examined using unimodal, multi-modal, fixed-dimensional multimodal, and composition benchmark functions. The obtained results and Wilcoxon signed-rank test show that MRA has the best ability to find the optimal solutions against other comparable optimizers. Moreover, four commonly used benchmark engineering challenges were used for more validation. The results also proved that the MRA can present superior performance compared to other meta-heuristic algorithms.

## REFERENCES

- [1] T.-Q. Wu, M. Yao, and J.-H. Yang, "Dolphin swarm algorithm," *Frontiers Inf. Technol. Electron. Eng.*, vol. 17, no. 8, pp. 717–729, 2016.
- [2] S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks*. Cham, Switzerland: Springer, 2019, pp. 43–55.
- [3] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.
- [4] D. Karaboga, "Artificial bee colony algorithm," *Scholarpedia*, vol. 5, no. 3, p. 6915, 2010.
- [5] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, Jun. 2007.
- [6] J. S. Lewis and W. W. Schroeder, "Mud plume feeding, a unique foraging behavior of the bottlenose dolphin in the Florida keys," *Gulf Mex. Sci.*, vol. 21, no. 1, p. 9, Jan. 2003.
- [7] W. Qiao and Z. Yang, "An improved dolphin swarm algorithm based on kernel fuzzy C-means in the application of solving the optimal problems of large-scale function," *IEEE Access*, vol. 8, pp. 2073–2089, 2020.
- [8] A. T. Kamil, H. M. Saleh, and I. H. Abd-Alla, "A multi-swarm structure for particle swarm optimization: Solving the welded beam design problem," *J. Phys.: Conf.*, vol. 1804, no. 1, Feb. 2021, Art. no. 012012.
- [9] M. Sarkar and T. K. Roy, "Optimization of welded beam structure using neutrosophic optimization technique: A comparative study," *Int. J. Fuzzy Syst.*, vol. 20, no. 3, pp. 847–860, Mar. 2018.
- [10] Y. Çelik and H. Kutucu, "Solving the tension/compression spring design problem by an improved firefly algorithm," *IDDM*, vol. 1, no. 2255, pp. 1–7, 2018.
- [11] F. Zitouni, S. Harous, and R. Maamri, "The solar system algorithm: A novel Metaheuristic method for global optimization," *IEEE Access*, vol. 9, pp. 4542–4565, 2021.
- [12] M. Durđev, E. Desnica, J. Pekez, M. Milošević, D. Lukić, B. Novaković, and L. Luka Dorđević, "Modern swarm-based algorithms for the tension/compression spring design optimization problem," *Ann. Fac. Eng. Hunedoara*, vol. 19, no. 2, pp. 55–58, 2021.
- [13] A. E. Yildirim and A. Karci, "Application of three bar truss problem among engineering design optimization problems using artificial atom algorithm," in *Proc. Int. Conf. Artif. Intell. Data Process. (IDAP)*, Sep. 2018, pp. 1–5.
- [14] H. Fauzi and U. Batool, "A three-bar truss design using single-solution simulated Kalman filter optimizer," *MEKATRONIKA*, vol. 1, no. 2, pp. 98–102, Jul. 2019.
- [15] S. Q. Salih, A. A. Alsewari, and Z. M. Yaseen, "Pressure vessel design simulation: Implementing of multi-swarm particle swarm optimization," in *Proc. 8th Int. Conf. Softw. Comput. Appl.*, Feb. 2019, pp. 120–124.
- [16] H. M. Mohammed and T. A. Rashid, "A novel hybrid GWO with WOA for global numerical optimization and solving pressure vessel design," 2020, *arXiv:2003.11894*.
- [17] G. Kiruthiga, S. Krishnapriya, V. Karpagambigai, N. Pazhaniraja, and P. V. Paul, "A novel bio-inspired algorithm based on the foraging behaviour of the bottlenose dolphin," in *Proc. Int. Conf. Comput. Power, Energy, Inf. Commun. (ICCPEIC)*, Apr. 2015, pp. 0209–0224, doi: 10.1109/ICCPEIC.2015.7259465.
- [18] B. Wursig and W. F. Perrin, *Encyclopedia of Marine Mammals*, 2nd ed. Amsterdam, The Netherlands: Elsevier, 2009, doi: 10.1016/B978-0-12-373553-9.X0001-6.
- [19] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.
- [20] J. G. Digenakis and K. G. Margaritis, "On benchmarking functions for genetic algorithms," *Int. J. Comput. Math.*, vol. 77, no. 4, pp. 481–506, Jan. 2001.
- [21] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 6, pp. 617–644, Dec. 2009.
- [22] J. J. Liang, P. N. Suganthan, and K. Deb, "Novel composition test functions for numerical global optimization," in *Proc. IEEE Swarm Intell. Symp. (SIS)*, Jun. 2005, pp. 68–75.
- [23] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on realparameter optimization," Kanpur Genet. Algorithms Lab., IIT Kanpur, Nanyang Technol. Univ., Singapore, KanGAL Rep. 2005005, May 2005.
- [24] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, "Honey badger algorithm: New Metaheuristic algorithm for solving optimization problems," *Math. Comput. Simul.*, vol. 192, pp. 84–110, Feb. 2022.
- [25] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowl.-Based Syst.*, vol. 191, Mar. 2020, Art. no. 105190.
- [26] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
- [27] S. Mirjalili, "The antlion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015.
- [28] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
- [29] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.
- [30] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci.*, 1995, pp. 4–39.
- [31] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*. Boca Raton, FL, USA: CRC Press, 2020.
- [32] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017.
- [33] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, Feb. 2016.
- [34] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.

- [35] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.
- [36] A. Sadollah, A. Bahreinejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2592–2612, May 2013.
- [37] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A Metaheuristic approach to solve structural optimization problems," *Eng. with Comput.*, vol. 29, no. 1, pp. 17–35, Jan. 2013.
- [38] T. Ray and P. Saini, "Engineering design optimization using a swarm with an intelligent information sharing among individuals," *Eng. Optim.*, vol. 33, no. 6, pp. 735–748, Aug. 2001.
- [39] A. Kaveh and A. Dadras, "A novel meta-heuristic optimization algorithm: Thermal exchange optimization," *Adv. Eng. Softw.*, vol. 110, pp. 69–84, Aug. 2017.
- [40] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, pp. 2232–2248, Jun. 2009.
- [41] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Eng. Appl. Artif. Intell.*, vol. 20, pp. 89–99, Feb. 2007.
- [42] E. Mezura-Montes and C. A. C. Coello, "An empirical study about the usefulness of evolution strategies to solve constrained optimization problems," *Int. J. Gen. Syst.* vol. 37, pp. 443–473, Aug. 2008.
- [43] A. Kaveh and M. Khayatizad, "A new meta-heuristic method: Ray optimization," *Comput. Struct.*, vol. 112, pp. 283–294, Dec. 2012.
- [44] C. A. C. Coello, "Constraint-handling using an evolutionary multiobjective optimization technique," *Civil Eng. Environ. Syst.*, vol. 17, no. 4, pp. 319–346, Oct. 2000.
- [45] L. D. S. Coelho, "Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems," *Expert Syst. Appl.*, vol. 37, no. 2, pp. 1676–1683, Mar. 2010.
- [46] Q. He and L. Wang, "A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization," *Appl. Math. Comput.*, vol. 186, no. 2, pp. 1407–1422, Mar. 2007.
- [47] H. H. Rosenbrock, "An automatic method for finding the greatest or least value of a function," *Comput. J.*, vol. 3, no. 3, pp. 175–184, Mar. 1960.
- [48] A. Kaveh and T. Bakhshpoori, "Water evaporation optimization: A novel physically inspired optimization algorithm," *Comput. Struct.*, vol. 167, pp. 69–85, Apr. 2016.
- [49] A. H. Gandomi, X.-S. Yang, A. H. Alavi, and S. Talatahari, "Bat algorithm for constrained optimization tasks," *Neural Comput. Appl.*, vol. 22, no. 6, pp. 1239–1255, May 2013.
- [50] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: Charged system search," *Acta Mechanica*, vol. 213, nos. 3–4, pp. 267–289, Sep. 2010.
- [51] K. S. Lee and Z. W. Geem, "A new structural optimization method based on the harmony search algorithm," *Comput. Struct.*, vol. 82, nos. 9–10, pp. 781–798, Apr. 2004.
- [52] F.-Z. Huang, L. Wang, and Q. He, "An effective co-evolutionary differential evolution for constrained optimization," *Appl. Math. Comput.*, vol. 186, no. 1, pp. 340–356, Mar. 2007.
- [53] K. M. Ragsdell and D. T. Phillips, "Optimal design of a class of welded structures using geometric programming," *J. Eng. Ind.*, vol. 98, no. 3, pp. 1021–1025, Aug. 1976, doi: [10.1115/1.3438995](https://doi.org/10.1115/1.3438995).
- [54] K. Deb, "Optimal design of a welded beam via genetic algorithms," *AIAA J.*, vol. 29, no. 11, pp. 2013–2015, Nov. 1991.



**MEHMET AKIF CIFCI** (Member, IEEE) received the B.A. degree from the Department of ELT, Dokuz Eylül University, the B.C.S. degree from the Department of Computer Engineering, UoP University, and the M.C.S. and Ph.D. degrees in computer science from Istanbul Aydin University. He was an educator with over ten-year experience and an active researcher. He has published several research articles, presented at conferences and has provided training sessions on image processing, NLP, food safety, and data analyses. His research interests include AI, ML, and DL. He is a member of CMC and a Founder Member of ABV.



**SAMINA KAUSAR** is currently pursuing the Ph.D. degree with the School of Computer Engineering and Science, Shanghai University, China. She is also working as a Researcher with the School of Computer Engineering and Science, Shanghai University. She is also working as an Assistant Professor at the University of Kotli Azad Jammu and Kashmir, Kotli Azad Kashmir, Pakistan. Her research interests include big data, bioinformatics, computer networks, cloud computing, data mining, e-learning, and machine learning algorithms.



**SADIQ HUSSAIN** received the Ph.D. degree from Dibrugarh University, Assam, India. He is currently a System Administrator at Dibrugarh University. He is associated with Computerization of the Examination System, Dibrugarh University. He had published various research and conference papers of international repute. His research interests include data mining, medical analytics, and machine learning. He had reviewed research papers for many IEEE, SCI, and Scopus indexed journals. He had acted as a technical member of many reputed conferences.



**LAMIAA M. EL BAKRAWY** received the M.S. and Ph.D. degrees in computer science from the Faculty of Science, Al-Azhar University, Egypt, in 2009 and 2012, respectively. Her master's topic was optimization using swarm intelligence. Her Ph.D. degree topic was machine learning in image authentication. She is currently an Associate Professor with the Mathematics and Computer Department, Faculty of Science, Al-Azhar University. She has published many articles in well-known international journals. Her research interests include image processing, social network analysis, computational intelligence, machine learning, metaheuristic optimization, and information security. She has reviewed research papers for many Scopus-indexed journals. She is also a program committee member in various international conferences.



**ABEER S. DESUKY** received the B.Sc. degree in science, in 2003, the M.Sc. degree in computer science, in 2008, and the Ph.D. degree in computer science, in 2012. She was an Associate Professor, in 2017. She is currently an Associate Professor in computer science at the Mathematics Department, Faculty of Science, Al-Azhar University, Cairo, Egypt. She has published a number of research papers in the field of AI, machine learning, metaheuristic optimization, and data mining and analysis. She is the supervisor of some master's and doctoral thesis. She is a Reviewer in many Scopus-indexed journals such as IEEE ACCESS, *Egyptian Informatics Journal*, and *PeerJ*.