Kurdistan Journal of Applied Research (KJAR)
Print-ISSN: 2411-7684 | Electronic-ISSN: 2411-7706

Original Article

Check for updates

A Performance-Driven Enhancement of the Mud Ring Algorithm for Global Optimization Challenges

Shabaz Kawa Ali [a, b *], **Azad Abdullah Ameen** [c]

[a] Department of Computer Science, College of Science, Charmo University, Sulaymaniyah, Iraq.
[b] Information Technology Department, Kurdistan Technical Institute, Sulaymaniyah, Iraq.
[c] Department of Software Engineering, College of Engineering and Computational Science, Charmo University, Sulaymaniyah, Iraq.

Submitted: 02 July 2025
Revised: 03 August 2025
Accepted: 20 August 2025

*** Corresponding Author**:
shabaz.ali@chu.edu.iq

Keywords: Metaheuristic, Optimization, MRA, Enhanced mud ring algorithm, Engineering design.

How to cite this paper: S. K. Ali, A. A. Ameen, "A Performance-Driven Enhancement of the Mud Ring Algorithm for Global Optimization Challenges", KJAR, vol. 10, no. 2, pp: 178-211, Dec 2025, doi: 10.24017/science.2025.2.13

Abstract: Nowadays, real-world optimization problems are becoming increasingly complex tasks, prompting the development of nature-inspired algorithms that mimic biological phenomena to improve search performance and solution quality. The Mud Ring Algorithm (MRA), inspired by the cooperative hunting behavior of bottlenose dolphins, has shown promise but remains sensitive to parameter settings, especially when balancing exploration and exploitation. To address these limitations, this paper proposes the Enhanced Mud Ring Algorithm (EMRA), which introduces a novel mechanism to more effectively manage the exploration-exploitation tradeoff. This modification allows the algorithm to escape local minima and explore the solution space more effectively. Numerical experiments on some standard benchmark functions as well as the difficult Congress on Evolutionary Computation 2019 benchmark suite show that EMRA outperforms the original MRA in terms of accuracy performance and computational cost, especially when dealing with high-dimension and multi-peak functions. In addition, EMRA was used in three complex engineering optimization problem designs (welded beam, pressure vessel, and tension spring), and the results were found to be more accurate and reliable than MRA. These results validate the strength and applicability of EMRA as a general optimization tool to tackle complex problems from many different disciplines, including real-world problems requiring the exhaustive exploration of multiple options. In summary, this study shows that EMRA is an effective extension of metaheuristic optimization that is applicable in real-world problems.

1. Introduction

Optimization is the process and result of achieving the most optimal way to solve a problem given a set of objectives and constraints. It is an important concept in many fields where one wishes to make a decision by selecting the best option, where each option has an associated utility or cost. The current optimization techniques, such as metaheuristic algorithms, are utilized for solving complex multi-dimensional problems, where classical mechanisms may not yield desirable results [1, 2]. Metaheuristics or nature-inspired optimization algorithms are a family of optimization algorithms based on principles in nature and natural phenomena such as living systems and hunting behaviors. Here, the focus is on the class of optimization algorithms that are based on the imitation of evolutionary process, swarm behaviors and principles of natural selection.

These methods, such as Genetic Algorithms with mutation and crossover operations and Particle Swarm Optimization (PSO) with the flocking and swirling behavior of birds or fish, operate based on evolution [1, 3, 4]. Metaheuristic approaches are well known due to their excellent flexibility and

robustness when solving hard optimization problems that do not need gradient information or the convexity condition. Previous studies [2, 5, 6] presented significant contributions in this area. These studies operated through an accurate balance between exploration (searching in the solution space) so as not to get stuck in local minima and exploitation (optimizing for the best solutions), so as to improve upon the best solutions found. This adaptive balance is typically achieved through two types of mechanism: one that promotes population diversity by managing trade-offs between good and poor solutions, and another that drives convergence by refining solutions. Additionally, many metaheuristics employ adaptive dynamics to occasionally accept worse solutions, allowing them to escape local optima and dynamically adjust their search strategy over time.

Due to their adaptability and simplicity, there has been widespread application across various fields, from engineering design to data analysis and machine learning [7]. In addition, with advancements in technology and computer science, significant changes have taken place in various fields, such as healthcare [8]. Among these methods and algorithms, Mud Ring Algorithm (MRA) has been selected, which is a recently proposed meta-heuristic optimization procedure. It was inspired by the bottlenose dolphin's mud ring foraging behaviour. Algorithms are utilized to imitate their collaborative foraging activities in two phases: global search (exploration) mode and local search (exploitation) mode, which is an efficient approach to address complex optimization problems [9]. This algorithm has proven to be effective at addressing a wide range of real-world problems and it has been successfully applied across multiple domains, exhibiting strong performance in various search spaces, as discussed in the related work section [10, 11].

The MRA has some limitations. For example, MRA is highly sensitive to the transition factor (K), which controls the balance between exploration and exploitation. If K is poorly tuned, it can result in premature convergence on the local optima or insufficient exploration of the solution space. This makes the algorithm reliant on the proper setting of K, which can limit its effectiveness across different optimization problems. To overcome this limitation in MRA, we introduce the Enhanced Mud Ring Algorithm (EMRA), which improves upon the original MRA to achieve superior performance in solving global optimization problems. The major benefits of this enhancement are as follows:

- The proposed algorithm enhances MRA by introducing a better way to strike a balance between exploitation and exploration, thereby enhancing search efficiency and optimization.
- EMRA is thoroughly evaluated using 23 standards benchmarks and CEC 2019 test functions, showing better performance than MRA and getting results that are competitive with other algorithms.
- To validate its performance, the algorithm was applied to three functional constrained engineering design problems, highlighting its strength in solving complex tasks.
- The algorithm's efficiency and robustness have been enhanced for complicated optimization tasks.

The rest of the paper is organized as follows: in section 2, we describe the MRA theory. Related work and some MRA applications in various domains are described in section 3. Section 4 discusses the EMRA proposed itself, as well as its improvements and modifications. We present a full performance vs. benchmark analysis study of EMRA in section 5, comparing it with MRA as well as popular optimization methods. EMRA is also applied to three engineering design problems (welded beam, pressure vessel and tension spring design) to demonstrate its capabilities in real-life optimization problems in section 6. Finally, section 7 summarizes the main findings and outlines directions for future research.

2. Related Works

This section gives an overview of MRA from three different points of view. First, it highlights recent research on optimization algorithms, emphasizing current trends and new developments. Second, it presents practical examples of MRA being applied in various fields, demonstrating its effectiveness in solving real-world optimization problems. Finally, it presents MRA in detail.

2.1. Optimization Algorithms: Trends and New Developments

Various algorithms have been deployed for optimization problems, such as the well-known PSO [12]. PSO is extensively used in a variety of domains and could even contribute to improving security in cloud-based healthcare systems. Irshad *et al*. [13] invented a multi-objective approach and Bee-Foraging Learning based PSO (BFL-PSO) to balance the hiding ratio, data modification, and content preservation. Their method, BFL-PSO, performs better in security and the convergence rate, and reduces delays and error rates. This new protocol was utilized to enhance cryptographic key generation for enhanced data cleansing and secure health data transfer, according to the organization. PSO is also effective in the operation of power systems. For instance, a modified PSO integrated with a Feedforward Neural Network was suggested by Chafi and Afrakhte [14]. Their hybrid model was applied to the tuning of parameters, yielding more accurate short-term load forecasts.

The Grey Wolf Optimizer (GWO) presented by Mirjalili *et al*. [15] in 2014 is based on the social and hunting behavior of grey wolves in the wild. In another example, Hu *et al*. [16] proposed an improved version of GWO, BGWO, for discrete-binary type optimization problems such as feature selection. In their experiments, the proposed BGWO outperformed a standard GWO and also achieved better results with a faster convergence, especially for problems with large and complex datasets.

Similarly, the Whale Optimization Algorithm (WOA), being inspired by the bubble-net hunting method of humpback whales proposed by Mirjalili and Lewis [17] , has shown its ability to solve serious optimization problems. Such a method makes it possible for the algorithm to search the space more fully and directly, and gains a solution with good performance. WOA has also been successfully used in other application areas, for instance, robotics. For example, Xu *et al*. [18] proposed a new hybridization of the two algorithms by proposing the algorithm IWOA-SA which combines IWOA and SA. They used this method to help a platoon of four robots to rapidly learn the fastest paths.

Different applications have employed the Sine Cosine Algorithm (SCA) [19]. A multi-strategy version called Enhanced Sine–Cosine Algorithm (ESCA) was recently proposed by Zhou and Shang to enhance the parameter identification in photovoltaic (PV) models. It was also more effective and flexible than conventional techniques such as PSO and Differential Evolution [20]. Additionally, Ragab *et al*. [21] employed SCA for tuning hyperparameters on a deep learning model for secure, privacy-preserving healthcare systems based on blockchain.

The Salp Swarm Algorithm (SSA), a swarm-based optimization algorithm recognized by Mirjalili *et al.* [22], mimics the aggregation of salp chains in the ocean. Despite its compatibility with both optimization and model training, SSA has only been proven in experimental results. Yang *et al*. [23] proposed a self-learning adaptation of SSA which modifies the search while running. It improved the ability of neural networks to learn from medical data as well as the algorithm to perform faster and better in relation to the original SSA.

The Tree-Seed Algorithm (TSA) created by Kiran in 2015 is based on how trees dispense their seeds. It has been successfully used subsequently in various applications [24]. TSA has also been applied to determine the key factors in photovoltaic models. For example, Beşkirli and Dağ [25] demonstrated that TSA was able to extract these values fast and accurately compared to other methods. TSA has also been employed to address numerical and engineering optimization challenges. For example, Liu *et al*. [26] enhanced TSA by introducing a new variant. In this release, pattern search, dimension permutation, and elimination update mechanisms are considered to improve convergence rate and solution precision in solving benchmark problems and real-world design problems. On a side note, the Harris Hawks Optimization (HHO) was initially devised by Mashaleh *et al.* [27]. It's based on how Harris hawks hunt together [6]. It has been effectively used for spam detection with an accuracy of 94.3%. Expanding on this, Akl *et al.* [28] proposed the algorithm of Improved Harris Hawks Optimization. They also provided an enhancement of this version, which improved both exploration and exploitation. In particular, they used different location-based habitats during exploration to guide the algorithm on diverse regions of the solution space. They also introduced logarithm and exponent operations to avoid the local minimum, which makes the algorithm more reliable overall. The Circle Search Algorithm (CSA) was proposed by Qais *et al*. [29] and uses circular trajectories to accelerate convergence in optimization problems. CSA has been applied successfully to different optimization

problems, especially for modeling the parameters in proton exchange membrane fuel cells (PEMFC). For instance, Qais *et al*. [30] proposed a modified version of the CSA in order to obtain a more precise electrical model of PEMFC which could determine the seven influential factors governing the dynamic behavior of PEMFCs. The last one is the Chaos Game Optimization (CGO), which was introduced by Talatahari and Azizi [31] employing both fractal patterns and randomness, which can contribute to more extensively searching the space and avoiding trapping. Thus far, CGO has achieved promising results in medical image classification. Mabrouk *et al*. [32] employed CGO, in combination with deep transfer learning in a healthcare Internet of Things (IoT) environment, to determine the most relevant features. They achieved high accuracy when classifying melanoma and leukemia images, and also reduced the computing time.

2.2. Applications of the Mud Ring Algorithm Across Diverse Domains

MRA has been extensively investigated by many research fields and is confirmed to be a flexible and efficient tool for handling difficult problems. It is particularly remarkable because it successfully balances between the exploitation of new possibilities and the exploitation of promising solutions, thereby being characterized by strong convergence properties [9]. There are also various works that have combined MRA with other optimization methods in order to improve its performance in solving different problems, for example MRA hybridized with the Elk Herd Optimizer. This section not only presents the latest applications of MRA but also its application in the analysis of feature selection, energy management, air quality prediction, health, and industrial problems. A detailed comparison of MRA applications in terms of usability, performance, and scalability is illustrated in table 1.

Table 1: Comparative analysis of MRA applications in terms of usability, performance, and scalability.

Ref.	Application	Problem & Setup	Performance	Scalability
[33]	Feature selection in medical datasets.	Applied on 13 real datasets, SVM-based survival models	Accuracy (Acc.) increases from 94.37% to 95.77%, F1 score increase to 94.55%	Up to 90 features, binary mask encoding
[10]	TEG MPPT Control.	Applied on 4-string TEG emulator under dynamic temperature, on board MCU	Acc. 99.95% power tracked vs 99.14% (PSO)	4 agents, runs on Arduino Atmega2560; real-time
[34]	Air Quality Forecasting.	Applied on 3 000 samples, 6 classes in dataset;	Acc. 99.26% vs 98.34% (XGBoost), F1 97.75% vs 97.14%	3 000 × (pollutants + weather) data
[35]	IoT Waste Routing Path.	Smart bins with real-time fill-level, optimize truck routes	Less distance to collect waste bins to center vs static routes	Scales to hundreds of bins, GPS-equipped trucks
[36]	ECG Disease Diagnosis.	Biomedical signal classification	Acc. Increase 4.2 pts (88.75→92.94 % avg), F1 score ~90–94 %	Preprocessed 3000 ECG records
[37]	Wireless Sensor Network Trust Routing.	100-node WSN with 50 malicious; optimize CH-based trust routing	Lifetime increase to 3050 rounds; PDR 0.98 vs 0.92–0.96	100 nodes; trust calculate (direct/indirect/recent)
[38]	Converter Modulation for wind generator.	3-phase to 6-phase matrix converter for wind generator, mod functions tuned	THD decrease to 2.2%	6-phase topology; modulation function search
[39]	Network-Anomaly Detection.	(DDoS) attacks on 11011 utilized records	Accuracy (Acc.) increases from 94.58% to 96.39% for intrusion detection	Population counts 10, Max. iteration 50
[40]	Maternal-Risk Prediction.	SVM and KNN parameter tuning by MRA, applied on 13 real datasets.	Prediction accuracy increased up to 17.08%	Crossover oversampling technique used

2.2.1.MRA in Feature Selection and Machine Learning

One of the most notable applications of MRA is for feature selection to improve the accuracy of high-dimensional dataset classification. In a study by Bakrawy *et al.* [33], an MRA-based feature selection approach was proposed to enhance the survival forecast of children with hematopoietic stem-cell transplantation. Their investigation demonstrated that MRA can achieve better performance than

traditional feature selection methods, as measured by the substantially decreased number of features on the one hand and good prediction performance on the other. Both methods improve classification work by reducing dimensionality with no degradation of performance. Detecting Distributed Denial of Service (DDoS) attacks in Internet of Things – Software Defined Networking (IoT-SDN) is challenging due to the high traffic volume from distributed devices, the resource constraints of IoT nodes, and the dynamic SDN topology. In this paper, they provide a solution to the problem of detecting DDoS attacks in IoT-SDN based networks by building a deep learning-based anomaly detection model. The Predefined Mud Ring Algorithm(P-MRA) is applied to extract relevant features from the Network Intrusion Detection by Information Security Centre of Excellence (IDS ISCX 2012) dataset that are essential for the correct detection of attack. P-MRA discards irrelevant data and enhances the performance of the Multi-Serial Stacked Network (Multi-SSN). The features selected improve learning, resulting in more accurate detection and less false positives. This application exemplifies the capability of MRA for feature-selection in cybersecurity that enables not only an improved detection performance but also an increased efficiency within sophisticated and resource-constrained IoT systems [39].

2.2.2.MRA in Renewable and Wind Energy Optimization

In renewable energy, an MRA-based Maximum Power Point Tracking (MPPT) controller for centralized thermoelectric generators (TEG) is presented by Zafar *et al*. [10], where dynamic thermal gradients are taken into account. The results revealed that MRA yielded fast convergence, local optima evading and high energy extraction efficiency enhancement. Unlike conventional MPPTs at the literature such as Perturb & Observe (P&O) and PSO, MRA presented better adaptability towards varying thermal environments. In this work, a hybrid MRA-(EHO) Elk Herd Optimizer is proposed to optimize the modulation functions to be applied in the Multiphase Matrix Converter for wind energy applications. It aims to seamlessly connect the six-phase induction generators to the three-phase grids. In this scenario, MRA's global search capability offers optimal modulation indices, whereas EHO performs local refinement. This results in a 2.2% lower Total Harmonic Distortion (THD) than with previous methods [38].

2.2.3. MRA in Air Quality Prediction

A recent study by Sivanesh *et al.* [34] applied MRA in an ensemble voting-based deep-learning model for air quality forecasting in intelligent transportation systems. The proposed model utilized Long Short-Term Memory, deep belief networks, and stacked autoencoders for improved air quality forecasting. MRA was employed for hyperparameter optimization, with improved prediction accuracy. Their findings show the promise of MRA in deep learning models for dimensionality reduction in environmental use.

2.2.4. MRA in Healthcare and Medical Diagnosis

MRA has been investigated in several studies for clinical work. In a study by Alluhaidan *et al.* [36]. MRA-auxiliary deep learning model was proposed for Electrocardiogram (ECG) monitoring and cardiovascular disease diagnosis. For feature extraction, the model adopted the Stacked Autoencoder Topographic Map and MRA was applied to optimize the hyperparameters in better classification accuracy. The results indicate that MRA has significantly improved disease detection by improving both the model convergence and stability. In the development of an intelligent diagnostic system for the prediction of maternal health risk, MRA technology has also been successfully applied. Studies that combine MRA with deep learning have shown much better computing speed, feature selection efficiency and disease classification accuracy. This article applies the MRA technique to improve mother health risk prediction accuracy, to optimize the hyperparameters of classifiers such as Support Vector Machine (SVM), Random Forest (RF) and K-Nearest Neighbor (KNN). In the first phase, MRA tunes the major parameters of the SVM model (MRA-SVM), then feeds it through multiple datasets for assessment. In the second phase, in order to solve the problem of data imbalance, the crossover oversampling approach is put to use before training classifiers are also used to optimize the parameters for

RF and KNN. The results show dramatic improvements in prediction accuracy. When combined with oversampling, MRA performance reaches up to 17.08%. This confirms MRA as being a robust optimizer to enhance machine learning models for healthcare prediction tasks [40].

2.2.5.MRA in Smart Waste Management and IoT

In the study Kona *et al.* [35], an efficient urban-focused smart waste management system that combines IoT, sensors, and optimization algorithms to automate waste collection is proposed. In this application, the MRA serves as a routing algorithm to propose the shortest route cost and therefore the fuel needed by the waste disposal trucks of a refuse collecting company during their daily rounds. MRA is a technique able to address the vehicle routing problem by devising the path, evaluating the conditions (e.g., bin fill level, vehicle capacity, travel time), and selecting the best route.

2.2.6.MRA in Wireless Sensor Network

The work by Maradona and Jaya [37] concentrates on improving the performance of a Wireless Sensor Network (WSN) by implementing the clustering and routing protocols in the most efficient manner by making use of the MRA algorithm. The overall goal of WSNs, in general, is to conserve energy and prolong the network lifetime by properly selecting efficient Cluster Heads and constructing energy-aware routing paths. MRA is adopted to address the cluster head election problem which considers the nodes' residual energy, distance and signal strength. When MRA is employed, it also demonstrates savings in energy, improvements in network stability and better performance. Given its flexibility, our approach is suitable for dynamic WSN environments were node status and data traffic are subject to frequent changes.

2.3. Mud Ring Algorithm

This sub-section provides a full explanation of the biological processes and behaviors that led to the development of MRA. It then presents how the algorithm works and its functioning principles, and sketches the steps in the pseudocode. The second subsection finishes with messy tables of all sub-phases of the algorithm and the corresponding equations that describe each sub-phase's operation.

2.3.1. Inspiration

The MRA is based on the foraging behavior and mud-ring feeding strategy method of bottlenose dolphins, making MRA a new nature inspired optimization technique. Presented in 2022, MRA works by using mud plumes to confuse and surround prey'– replicating the hunting tactics of dolphins [9]. Similar to other population-based optimization methods, MRA randomly initializes a population of candidate solutions without prior knowledge of the optimal answer. Its intuitive method, efficient exploration and exploitation balance it, and its competitive convergence traits differentiate it from other optimization algorithms. However, MRA also faces potential challenges, particularly regarding its performance in high-dimensional and complex optimization problems, which have prompted ongoing research into algorithmic enhancements and improvements, especially balancing between exploration and exploitation [9, 33]. The detailed pseudocode of the MRA is shown in algorithm 1.

Algorithm 1: Mud ring algorithm.

1. Initialize a population of dolphins randomly D_i, $for\ i = 1,2,3, \ldots, n$ along with their velocities v_i
2. Evaluate the fitness value for each dolphin
3. Identify the best-performing dolphin position D^*
4. ***while*** $(t < T_{max})$
5. ***for*** $i = 1\ to\ n$
6. Update the control parameters $K, C, a, and\ l$
7. $if\ |K| \geq 1$:
8. - Create a new solution by adjusting velocity v_i using (Eq.3)
9. $else$:
10. - Simulate the "mud ring" behavior
11. - Update the dolphin's location based on the cooperative mechanism (Eq.5)
12. $end\ if$
13. ***end for***
14. Handle boundary constraints for dolphins outside the search limits
15. Recalculate the fitness value for each dolphin
16. Update D^*, if a better solution is found
17. Increment iteration counter: $t = t + 1$
18. ***end while***
19. Return D^* as the optimal solution

2.3.2. The MRA Algorithm Phases

The procedural framework of the MRA consists of several key phases, which are outlined below. The following sections provide a detailed discussion of each step:

- ***Searching for Prey (Exploration)***

The exploration phase simulates dolphins employing echolocation to randomly search for prey. Dolphins adjust their positions based on sound loudness, which decreases as they approach prey, represented mathematically as follows:

$$K = 2a.r - a \quad (1)$$

where (a) is calculated by: $a = 2\left(1 - \frac{t}{T_{max}}\right)$ (2)

Here, (K) sets a balance of exploration to exploitation, (r) is a random vector in the range between 0 and 1, and (t) represents the current iteration number. Also, (a) is a control variable used inside the MRA algorithm's loop. The dolphin's positions are updated using velocities, given by:

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

Where $(D^{\rightarrow t})$ denotes the dolphin's position at iteration (t), and V^t is the dolphin's velocity vector. This movement allows the dolphins to discover the search space in a diverse and wide-reaching manner during the early stages of the algorithm [9].

- ***Mud Ring Feeding (Exploitation)***

Once prey is detected, dolphins switch to the exploitation phase by forming a mud ring around the best-known prey location. The mathematical representation of this behavior is:

$$D^{\rightarrow t} = D^{\rightarrow *t-1} * \sin(2\pi * l) - K^{\rightarrow} * A^{\rightarrow} \quad (4)$$

where $(D^{\rightarrow *t-1})$ is the position of the most remarkable dolphin identified so far, (l) is a random number [0,1], and (A) is defined as:

$$A^{\rightarrow} = |C^{\rightarrow} * D^{\rightarrow *t-1} - D^{\rightarrow t-1}| \quad (5)$$

The coefficient (C) is calculated as: $C = 2 * r$ (6)

where (r) is again a random-vector between (0 and 1). These parameters ensure that the dolphins efficiently encircle and approach the optimal solution, maintaining a balance between local search (exploitation) and global search (exploration) [9].

3. Materials and Methods

The EMRA's goals are to enhance the novel MRA's performance by dynamically balancing the local search (exploration) and global search (exploitation) phases. In the original MRA, the key parameter (K), which controls this balance, is calculated by a static function dependent on a single linear parameter (a). In EMRA, the parameter (K) is calculated in three distinct phases according to the progression of (t), as shown in functions No. (7, 8, and 9):

Early Exploration Phase: if $\left(\frac{t}{T_{max}}\right) < 0.3$, then

$$K = \left((2ar - a) * \left(1 - \frac{t}{T_{max}}\right) + 0.1 * randn(1, \dim)\right) \quad (7)$$

- Here, $(r)$is a random-number in [0,1], and $(randn)$ introduces a small Gaussian noise and this phase encourages strong exploration by amplifying randomness and allowing wider search space coverage.

Balanced Phase: if $0.3 < \left(\frac{t}{T_{max}}\right) < 0.7$, then

$$K = (2ar - a) * exp^{-\left(\frac{t}{\frac{T_{max}}{10}}\right)} \quad (8)$$

- The exponential decay reduces the impact of randomness gradually and this phase provides a balance between exploration and then exploitation, focusing the search around promising areas without losing diversity. By doing so, the algorithm can explore new areas while refining known good solutions, enhancing its ability to increase the probability of discovering the global optimum and avoid local optima.

Exploitation Phase: if $\left(\frac{t}{T_{max}}\right) \geq 0.7$

$$K = 0.5 * (2ar - a) \quad (9)$$

- The parameter is scaled down to concentrate on an intensive local search around the best-known solutions and this phase accelerates convergence and refines solutions with reduced randomness.

In addition, EMRA implements a penalty function for managing constraints. Solutions that exceed variable limits or other problem limitations suffer a penalty equivalent to the level of the violation, like function no. (10).

$$penalty = \sum_{j=1}^{dim}(max\ (0, x_j - ub))^2 + (max\ (0, lb - x_j))^2) \quad (10)$$

- (lb) and (ub) are used to keep the search within valid limits for each variable in the optimization problem and (x_j) is used to specify the positions of each agent.

During the fitness evaluation, this penalty is added to the objective function value to discourage infeasible solutions and guide the search within valid regions, so at each iteration, the dolphin positions

are updated according to the adaptive (K), bounded within the search space, and evaluated for fitness with penalties applied as needed. The algorithm continues until reaching (T_{max}), ultimately returning the best feasible solution found. These modifications improve the algorithm's search balance and constraint handling, resulting in better optimization performance than the original. A detailed pseudo-code and flowchart representation of EMRA is shown in figure 1 and algorithm 2.

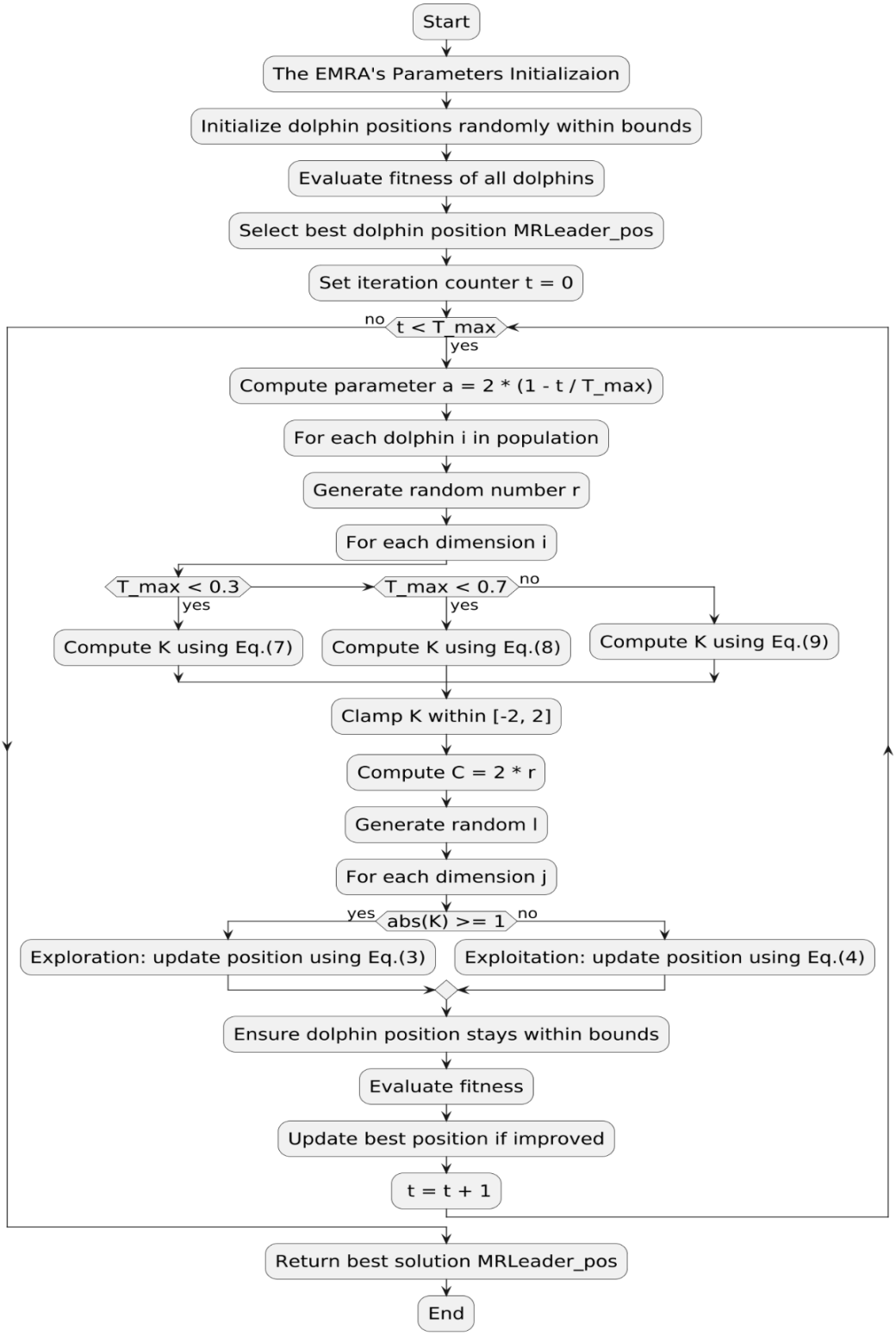

Figure 1: Flowchart illustration of EMRA.

Algorithm 2: Enhanced mud ring algorithm.

1. Initialize a population of dolphins randomly D_i, $for\ i = 1,2,3,\ldots.,n$ along with their velocities v_i
2. Compute penalty using Eq. (10) and evaluate the fitness value for each dolphin
3. Identify the best-performing dolphin position D^*
4. $\boldsymbol{while}\ (\boldsymbol{t} < T_{max})$
5. Compute parameter a, using Eq. (2)
6. $\boldsymbol{for}\ i = 1\ to\ n$
7. $if\ T_{max} < 0.3$
8. Compute parameter K using Eq. (7)
9. $elseif\ T_{max} < 0.7$
10. Compute parameter K using Eq. (8)
11. $else$:
12. Compute parameter K using Eq. (9)
13. end
14. Update the control parameters $K, C, and\ l$
15. $if\ |K| \geq 1$:
16. - Create a new solution by adjusting velocity v_i using (Eq.3)
17. $else$:
18. - Update the dolphin's location based on the cooperative mechanism (Eq.4)
19. $end\ if$
20. $\boldsymbol{end\ for}$
21. Handle boundary constraints for dolphins outside the search limits
22. Recalculate the fitness value for each dolphin
23. Update D^*, if a better solution is found
24. Increment iteration counter: $t = t + 1$
25. $\boldsymbol{end\ while}$
26. Return D^* as the optimal solution

4. Results

4.1. Experimental Configuration and Evaluation Approaches

All benchmark experiments were conducted using MATLAB R2023b on a Windows 11 (64-bit) based machine with 8 GB of RAM and an Intel Core i5-1334U 1.30GHZ CPU. To get reliable results, the first population was created at random for each execution. Both methods ran for 500 iterations with a population of 30 and all test runs were repeated 30 times to ensure that the experimental data was statistically significant. We also used MATLAB implementations of nine well-known, state-of-the-art optimization algorithms for a comprehensive comparative evaluation, specifically of CSA, SCA, SSA, HHO, WOA, PSO, TSA, GWO and CGO to perform performance comparisons with EMRA. The evaluation was conducted in such a way that the performance analysis was as comprehensive as possible based on two bench mark test set,s one with 23 standards and another with 10 CEC-2019 functions. Several important aspects were used if the EMRA had been carried out:

- Mean and standard deviation: This was used to show the average result and how stable the algorithm is across multiple runs.
- Statistical significance Test: Wilcoxon rank-sum and T-tests were used to check whether the differences between algorithms were meaningful (p-value < 0.05).
- Fair Initialization: All algorithms were tested with the same randomly generated starting population to ensure a fair comparison.
- Graphical Representation: Box and whisker plots were used to show how the performance results vary across different runs.
- Friedman Mean Rank Method: This method ranks algorithms across many functions and helps to compare their overall performance.

4.2. Comparative Analysis of EMRA with Original MRA Using Standard Benchmark Functions

In terms of the optimization problems, a set of 23 standard benchmark functions were taken for comparison. The functions were divided into three groups based on their nature: 1) unimodal (F1-F7), 2) multimodal (F8-F13) and 3) fixed-dimension multimodal (F14-F23).

4.2.1. Evaluation Using Uni-modal Functions (F1–F7)

The average (Avg.) and standard deviation (Std.) metrics were used to measure the accuracy and consistency of EMRA in identifying optimal solutions within smooth search spaces, as shown in table 2 (Table A1 in the appendix provides extra details). Despite consistently achieving optimal results similar to MRA in functions F1, F2, F3, and F4, EMRA effectively demonstrates its reliability and efficiency across these benchmark functions, never producing the worst results in terms of average performance or standard deviation. In F7, we obtained a lower result (8.402E-05) compared to the original (9.004E-05), indicating an improvement in performance. Regarding standard deviations representing the consistency and stability of the algorithm, EMRA performs better than MRA in two out of seven functions. In F5, EMRA shows better stability with a standard deviation of (5.241E+00), which is slower than MRA's (1.291E+01). Among the other functions, F7 also showed a similar improvement with EMRA showing lower stability with a standard deviation of (6.971E-05) compared to MRA (8.957E-05). MRA has a better consistency in F6 with a standard deviation (8.396E-07), although some variance is observed.

Table 2: Performance of EMRA compared to the original MRA on unimodal functions (F1 to F7).

Fun.	EMRA		MRA	
	Avg.	Std.	Avg.	Std.
F1	0	0	0	0
F2	0	0	0	0
F3	0	0	0	0
F4	0	0	0	0
F5	2.775E+01	5.241E+00	2.105E+01	1.291E+01
F6	3.811E-06	7.029E-06	4.788E-07	8.396E-07
F7	8.402E-05	6.971E-05	9.004E-05	8.957E-05

4.2.2. Evaluation Using Multimodal Functions (F8–F13)

Using the same evaluation metrics, EMRA outperforms MRA in three out of six multimodal functions, demonstrating its capability to explore complex landscapes and locate the global optimum as presented in table 3 (Table A2 in the appendix provides more details). Regarding the average, in F8, EMRA achieves a lower average (-9.102E+03) compared to MRA (-6.620E+03), highlighting its effectiveness in escaping local optima. Similarly, in F12, EMRA attains a lower average (3.829E-10) compared to MRA (6.012E-09). In F13, EMRA achieves a lower average (7.378E-09) compared to MRA's (8.358E-08). Nevertheless, in F9 and F11, the algorithms reach the best result of 0, which means that they perform equally well. However, EMRA still shows a better exploration process for most multimodal functions.

In terms of the standard deviation, EMRA seems to be more robust other than in two of the multimodal functions among the six. In F12, EMRA achieves higher stability with a smaller standard deviation value of 7.340E-10 than MRA's (1.368E-08), for a more stable performance across runs. The same is observed for EMRA (denoted F13), where EMRA beats MRA in consistency by having a standard deviation of (1.560E-08), which is far less than MRA's (1.383E-07). Additionally, in F9 and F11, EMRA obtained the same result for the standard deviations, which indicated no change in variance. These findings emphasize EMRA's enhanced stability and accuracy in solving multimodal functions.

Table 3: Performance of EMRA with original MRA on Multi-modal functions (F8 to F13).

Fun.	EMRA		MRA	
	Avg.	Std.	Avg.	Std.
F8	-9.102E+03	2.169E+03	-6.620E+03	1.990E+03
F9	0	0	0	0
F10	4.441E-16	3.009E-31	4.441E-16	3.009E-31
F11	0	0	0	0
F12	3.829E-10	7.340E-10	6.012E-09	1.368E-08
F13	7.378E-09	1.560E-08	8.358E-08	1.383E-07

4.2.3. *Evaluation Using Fixed-Dimension Multimodal Functions (F14–F23)*

This group of functions is particularly valuable for testing the balance between exploration and exploitation in optimization algorithms. As shown in table 4 (Table A3 in the appendix provides extra details about the functions), with respect to performance, EMRA demonstrates a significant improvement over MRA in fixed-dimension multimodal functions. In these functions, EMRA outperforms MRA in six out of ten, showcasing its superior optimization capability. For instance, significant contributions are made in F14, F15, and F16. In F14 and F15, EMRA achieves smaller averages of (1.387E+00) and (7.692E-04) compared to MRA's (4.111E+00) and (1.006E-03), respectively. Especially in F16, EMRA attains a lower average of (-9.652E-01) compared to MRA (1.027E+00), highlighting its superior optimization performance.

Further highlighting its capability in exploration, in this group of benchmark functions, significant contributions are made in F21, F22, and F23. EMRA achieves lower averages of (-7.604E+00), (-7.214E+00), and (-7.291E+00) in F21, F22, and F23, respectively, compared to MRA's (-1.015E+01), (-1.040E+01) and (-1.054E+01), improving the way it works by addressing complex fixed-dimension multimodal optimization challenges.

Regarding standard deviation, EMRA has greater consistency in F14, demonstrating enhanced reliability with a standard deviation of (2.131E+00), surpassing MRA's (5.250E+00). In F15, EMRA's standard deviation is (2.880E-04), which is lower than MRA's (3.831E-04). This trend continues in F16 with a lower standard deviation (5.774E-02) compared to MRA's (5.113E-03). In F19 and F20, EMRA's achievement of lower values indicates a more stable and reliable performance across multiple runs. In F19, EMRA demonstrates improved stability by achieving a lower standard deviation (6.110E-02), indicating more consistent performance across multiple runs compared to MRA's (6.786E-02). Similarly, in F20, EMRA outperforms MRA in terms of reliability, with a lower standard deviation (1.788E-01) compared to MRA's (2.059E-01), strengthening its capability to deliver consistent outcomes in challenging optimization landscapes.

Table 4: Performance of EMRA with original MRA in fixed-dimension multimodal functions (F14 to F23).

Fun.	EMRA		MRA	
	Avg.	Std.	Avg.	Std.
F14	1.387E+00	2.131E+00	4.111E+00	5.250E+00
F15	7.692E-04	2.880E-04	1.006E-03	3.831E-04
F16	-9.652E-01	5.774E-02	-1.027E+00	5.113E-03
F17	4.299E-01	2.629E-02	4.252E-01	2.078E-02
F18	1.276E+01	8.843E+00	8.366E+00	5.650E+00
F19	-3.735E+00	6.110E-02	-3.734E+00	6.786E-02
F20	-2.686E+00	1.788E-01	-2.704E+00	2.059E-01
F21	-7.604E+00	2.592E+00	-1.015E+01	1.199E-03
F22	-7.214E+00	2.648E+00	-1.040E+01	2.816E-03
F23	-7.291E+00	2.694E+00	-1.054E+01	1.566E-03

4.3. *Comparative Analysis of EMRA with MRA Using CEC2019 Benchmark Functions*

Regarding performance, as presented in table 5 (Table A4 in the appendix provides information about the functions), EMRA outperformed MRA in six out of ten functions, highlighting its effectiveness in tackling complex optimization challenges. In F3, EMRA exhibits a lower average (1.3703E+01) compared to MRA's (1.3705E+01), which indicates its superior ability to move through complex landscapes and avoid local optima. In F4, EMRA outperforms MRA with a mean of (1.657E+04) versus (3.027E+04), showing its efficiency in locating the global optimum for challenging optimization problems. For F5, EMRA is better with an average of (6.455E+00) compared to MRA's (8.261E+00), which further validates its optimization capability. In F7, EMRA continues to dominate by achieving a lower average (1.201E+03) than MRA's (1.903E+03), highlighting its stability and efficiency across multiple iterations. In F8, EMRA outperforms MRA with a mean of (7.326E+00) over (7.875E+00), demonstrating its ability to handle complex multimodal functions efficiently. Finally, in F9, EMRA

continues to perform well, averaging (2.496E+03) compared to MRA's (4.510E+03), indicating its superiority in tackling CEC-2019 benchmark functions.

In terms of the standard deviations, EMRA achieves better consistency in eight out of the ten functions, exhibiting a zero-standard deviation in F1, indicating perfect stability. From F2 to F10, EMRA achieves lower standard deviation values than MRA in all functions except F9, indicating more stable and reliable behavior across the runs. For instance, in F2, EMRA records (3.651E+03), outperforming MRA's (4.696E+03). Similarly, in F5, EMRA achieves (8.004E-01), which is significantly better than MRA's (1.071E+00), confirming better consistency. However, in F9, MRA slightly surpasses EMRA in terms of stability with a standard deviation of (5.494E+02) compared to EMRA's (7.041E+02). Overall, these results affirm that EMRA offers more consistent optimization behavior in the majority of CEC-2019 benchmark functions.

Table 5: Performance comparison of EMRA and MRA on CEC2019 benchmark functions.

Fun.	EMRA		MRA	
	Avg.	Std.	Avg.	Std.
F1	1.000E+00	0	1.000E+00	0
F2	1.595E+04	3.651E+03	1.642E+04	4.696E+03
F3	1.3703E+01	5.395E-04	1.3705E+01	1.068E-03
F4	1.657E+04	5.100E+03	3.027E+04	1.027E+04
F5	6.455E+00	8.004E-01	8.261E+00	1.071E+00
F6	1.342E+01	7.876E-01	1.299E+01	8.024E-01
F7	1.201E+03	1.553E+02	1.903E+03	3.447E+02
F8	7.326E+00	3.213E-01	7.875E+00	3.459E-01
F9	2.496E+03	7.041E+02	4.510E+03	5.494E+02
F10	2.178E+01	1.095E-01	2.164E+01	1.173E-01

4.4. Statistical Evaluation

There are clear differences in performance between the original MRA and the enhanced MRA, relative to the 23 standard benchmark functions. Table 6 shows that EMRA outperforms MRA across a number of functions, with particular gains in F8, F12, F13, F14, F15 and F16 where the p-value is less than < (0.05), indicating significant progress and a dash (–) in this table denotes values that are not available. For instance, in F8, EMRA gets an average of (-9.102E+03), while MRA gets an average of (-6.620E+03). The p-value, which is equal to (2.198E-05), shows a statistically significant enhancement. In F12 and F13, EMRA outperforms MRA with p-values of (2.825E-02) and (3.979E-03). This proves that EMRA is more efficient at solving these functions. In addition, F14, F15, and F16, EMRA yields better outcomes in these optimization goods with p values (1.084E-02), (8.982E-03), and (2.806E-07), respectively.

Moreover, in F21, F22, and F23, EMRA was able to outperform MRA with p-values of 1.375E-06, 1.415E-08, and 1.403E-08, respectively. In other words, EMRA was the unquestionable superior method in more intricate multidimensional optimization problems. EMRA also performs better in F18 with a p-value of 2.534E-02, which goes to show how effective it is. However, several functions, including (F1, F2, F3, F4, F9, F10, F11, F17, F19, and F20), demonstrate p-values beyond (0.05), indicating an absence of statistically significant differences between EMRA and MRA. In F1, F2, and F3, both algorithms yield identical outcomes and there is no variation, signifying no performance disparity. In F19 and F20, p-values of 9.252E-01 and 7.184E-01, respectively, indicate that EMRA's changes did not produce significant enhancements compared to MRA for these functions.

The evaluation of MRA and EMRA using the CEC-2019 benchmark functions highlights significant performance differences. As shown in Table 5, EMRA demonstrates statistically significant improvements over MRA in multiple functions, particularly in (F3, F4, F5, F6, F7, F8, F9, and F10) where the p-values are below 0.05, indicating notable enhancements. For instance, in F3, EMRA achieves an average of (1.3703E+01) compared to MRA's (1.3705E+01), with a p-value of (1.206E-10), confirming a statistically significant difference. Similarly, in F4, EMRA outperforms MRA with an average of (1.657E+04) versus (3.027E+04), and a p-value of (1.702E-08), highlighting meaningful improvement. In

functions like F5 and F6, EMRA also demonstrates superior performance, achieving p-values of 6.366E-10 and 3.926E-02, respectively, suggesting a strong advantage over MRA. Furthermore, in F7, F8, and F9, EMRA maintains its superiority with p-values of 1.604E-14, 3.375E-08, and 7.145E-18, respectively, reinforcing its effectiveness in handling these optimization problems. Additionally, in F10, EMRA achieves an average of (2.178E+01) compared to MRA's (2.164E+01), with a p-value of (2.074E-05), confirming its significant edge.

On the other hand, F1 and F2 exhibit p-values above > (0.05), demonstrating no statistically significant difference between EMRA and MRA. In F1, both algorithms yield identical results (1.00E+00) and signify no variation in performance. Similarly, in F2, the p-value of 6.665E-01 suggests that the performance difference between EMRA and MRA is not statistically significant. According to the results, EMRA performs much better than MRA in several test functions, with statistically significant improvements in key optimization tasks.

Table 6 : The p-values of the comparison between EMRA and MRA across standard benchmark functions and the CEC-2019 suite.

Standard Benchmark Functions				CEC-2019 Functions	
Function	**p-value**	**Function**	**p-value**	**Function**	**p-value**
F1	–	F12	2.825E-02	F1	–
F2	–	F13	3.979E-03	F2	–
F3	–	F14	1.084E-02	F3	1.206E-10
F4	–	F15	8.982E-03	F4	1.702E-08
F5	–	F16	–	F5	6.366E-10
F6	–	F17	–	F6	–
F7	–	F18	–	F7	1.604E-14
F8	2.198E-05	F19	–	F8	3.375E-08
F9	–	F20	–	F9	7.145E-18
F10	–	F21	–	F10	–
F11	–	F22	–		
		F23	–		

4.5. Convergence Behavior Analysis of EMRA against MRA

The convergence characteristics of EMRA and MRA were evaluated using CEC-2019 benchmark functions F1 to F10, as illustrated in figure 2. In functions F3, F4, F5, F7, F8, and F9, EMRA demonstrates a significantly faster and smoother convergence trend compared to MRA. In these cases, EMRA not only reaches the optimum in fewer iterations but also achieves lower fitness values, indicating better optimization performance. In functions F1 and F2, both algorithms converge to the same optimal value from the beginning, suggesting equivalent performance. However, in F6, MRA shows slightly better convergence in the later iterations, surpassing EMRA's performance as it continues to improve while EMRA plateaus. Overall, EMRA outperforms MRA in seven out of ten functions, highlighting its stronger convergence speed and solution quality across most test cases.

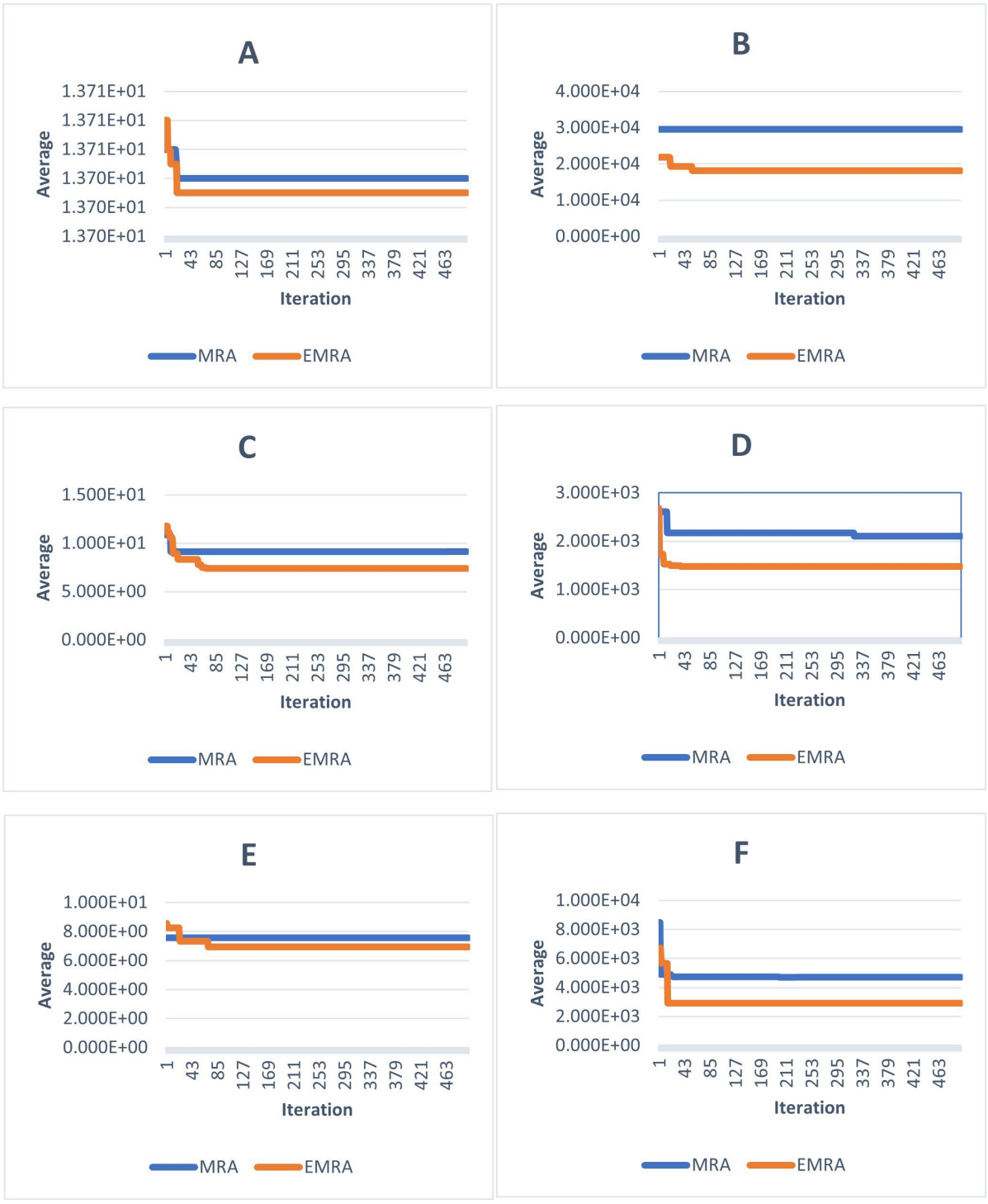

Figure 2: Convergence behavior of EMRA and MRA over iterations for the CEC2019 functions, where (A) corresponds to F3, (B) to F4, (C) to F5, (D) to F7, (E) to F8, and (F) to F9.

4.6. Evaluating the EMRA with Meta-heuristic Algorithms Using Standard Benchmark Functions

The effectiveness of the EMRA algorithm was evaluated by comparing it with nine recently developed and widely recognized metaheuristic algorithms across 23 standard benchmark functions. These include CSA, SCA, SSA, HHO, WOA, PSO, TSA, GWO and CGO. The goal of this comparison is to demonstrate the performance capabilities of EMRA. This comparison study wants to show not just EMRA's effectiveness but also its strength, competitiveness, and ability to solve several optimization challenges with different degrees of complexity.

4.6.1. Unimodal Standard Benchmark Functions (F1–F7)

An evaluation of unimodal functions (F1–F7), as shown in table 7, reveals key performance differences between EMRA and nine competing metaheuristic algorithms. EMRA outperforms the other algorithms in five out of the seven functions, particularly in F1, F2, F3, F4 and F7, where it achieves a perfect average of (0.00E+00) with zero average performance in F1 to F4. This highlights EMRA's strong ability to solve simpler unimodal functions with high precision. However, in F5, EMRA does not perform as well, achieving an average of (2.78E+01), while other algorithms such as CSA achieve (0.00E+00). Similarly, in F6, EMRA has an average of (3.81E-06), outperforming some algorithms but still lagging behind the top-ranking ones. Conversely, in F7, EMRA achieves a perfect average of (8.40E-05), demonstrating its superior exploitation ability in this function.

The standard deviation values indicate the stability of EMRA's performance. In F1 to F4, EMRA maintains an optimal standard deviation of (0.00E+00), reflecting absolute consistency across the runs. However, in F5, EMRA records a standard deviation of (5.24E+00), suggesting fluctuations in the results, whereas CSA and HHO maintain more stable performances. In F6, EMRA achieves (7.03E-06), outperforming WOA (1.44E-02) but being less stable than CSA (0.00E+00). In F7, EMRA demonstrates good stability, with a standard deviation of (6.97E-05), which is lower than CSA (7.10E-04) and SCA (5.99E-04), confirming its reliability in this function. Compared to other algorithms, EMRA shows a balance between consistency and performance, although it does not always achieve the lowest variability.

The p-value analysis highlights EMRA's statistical differences compared to other algorithms. In F1 to F4, EMRA achieves optimal values with no variance in p-values, indicating no difference from high-performing algorithms like CSA. In F5, the p-value of (6.47E-13) suggests EMRA underperforms compared to CSA and CGO. However, in F6 with a p-value of (5.81E-03), EMRA performs significantly better and differently from weaker algorithms like GWO and WOA. The strongest significance appears in F7, where a p-value of (3.77E-09) confirms EMRA's superior performance over SSA, SCA, and HHO. While EMRA excels in some functions, areas such as F5 require improvement to achieve greater statistical dominance.

Among the compared algorithms, EMRA demonstrated outstanding performance, achieving the top rank in most unimodal benchmark functions, notably ranking first for functions F1 through F4 and F7. Others, such as CSA and CGO, also showed competitive performance, frequently occupying top-tier rankings. Conversely, algorithms such as PSO and TSA showed comparatively weaker results, often ranking in lower positions across multiple functions. These rankings clearly illustrate variations in performance, underscoring EMRA's robustness and effectiveness relative to other state of the art optimization algorithms in diverse problem scenarios.

Table 7: Statistical comparison of EMRA and other metaheuristic algorithms in unimodal benchmark functions (F1–F7)

Algorithm	Function/ Metric	F1	F2	F3	F4	F5	F6	F7
EMRA	Average.	0	0	0	0	2.78E+01	3.81E-06	8.40E-05
	Std.	0	0	0	0	5.24E+00	7.03E-06	6.97E-05
	Ranking	1	1	1	1	9	6	1
CSA	Average.	3.85E-235	5.89E-116	5.48E-213	1.22E-119	0	0	4.35E-04
	Std.	0	2.64E-115	0	6.66E-119	0	0	7.10E-04
	p-value	−	2.28E-01	−	3.20E-01	3.26E-36	4.33E-03	9.27E-03
	Ranking	2	2	2	2	1	1	4
SCA	Average.	2.77E-10	2.55E-08	5.59E-08	1.67E+00	6.20E+01	4.71E+00	4.53E-04
	Std.	8.17E-10	3.99E-08	1.34E-07	2.28E+00	1.33E+02	1.69E+00	5.99E-04
	p-value	6.81E-02	8.94E-04	2.602E-02	1.66E-04	1.65E-01	5.42E-22	1.41E-03
	Ranking	7	7	5	9	10	10	5
SSA	Average.	2.67E-07	1.41E-01	6.01E+02	1.06E+00	1.29E+01	1.69E-07	3.62E-02
	Std.	6.20E-07	1.54E-01	1.06E+03	1.00E+00	1.45E+01	1.88E-07	2.85E-02
	p-value	2.16E-02	5.19E-06	2.98E-03	3.07E-07	2.05E-06	6.27E-03	3.77E-09
	Ranking	8	10	9	8	7	3	9
HHO	Average.	6.76E-104	2.13E-50	1.88E-75	1.02E-106	9.80E-07	2.52E-06	1.34E-04
	Std.	3.38E-103	1.15E-49	1.03E-74	5.58E-106	5.28E-06	9.41E-06	1.26E-04
	p-value	2.77E-01	3.13E-01	3.20E-01	3.21E-01	3.26E-36	5.49E-01	6.23E-02
	Ranking	4	5	4	3	2	5	2
WOA	Average.	1.46E-75	2.49E-51	5.82E-01	3.08E-02	2.96E+00	7.55E-03	1.27E-03
	Std.	7.80E-75	9.38E-51	2.59E+00	7.53E-02	8.52E+00	1.44E-02	1.55E-03
	p-value	3.10E-01	1.52E-01	2.23E-01	2.90E-02	1.16E-19	5.81E-03	9.34E-05
	Ranking	5	4	7	5	3	7	6
PSO	Average.	1.31E-06	4.19E-04	1.10E+03	3.11E+00	1.10E+01	4.70E-07	3.93E-02
	Std.	4.52E-06	1.20E-03	1.15E+03	2.09E+00	1.32E+01	6.34E-07	2.58E-02
	p-value	1.19E-01	6.17E-02	2.63E-06	3.27E-11	2.27E-08	0.00E+00	1.75E-11
	Ranking	9	9	10	10	6	4	10
TSA	Average.	1.08E-04	1.40E-05	1.95E+02	8.66E-01	6.17E+00	6.31E-01	3.63E-03
	Std.	2.20E-04	2.62E-05	5.72E+02	8.29E-01	8.26E+00	7.69E-01	3.66E-03
	p-value	9.63E-03	4.90E-03	6.67E-02	3.86E-07	1.77E-17	0.00E+00	1.93E-06
	Ranking	10	8	8	6	4	9	8
GWO	Average.	8.35E-38	1.24E-22	1.13E-07	1.01E+00	1.63E+01	4.40E-01	1.51E-03
	Std.	1.79E-37	1.59E-22	3.67E-07	1.77E+00	1.56E+01	3.69E-01	9.66E-04
	p-value	1.32E-02	7.13E-05	9.61E-02	2.93E-03	3.38E-04	1.80E-08	4.56E-11
	Ranking	6	6	6	7	8	8	7
CGO	Average.	8.52E-137	1.14E-71	7.67E-96	6.38E-58	1.09E+01	2.09E-17	3.37E-04
	Std.	3.72E-136	2.30E-71	3.68E-95	2.33E-57	8.55E+00	8.20E-17	2.10E-04
	p-value	2.16E-01	8.91E-03	2.58E-01	1.40E-01	6.47E-13	4.33E-03	4.90E-08
	Ranking	3	3	3	4	5	2	3

4.6.2. Multimodal Standard Benchmark Functions (F8–F13)

In an evaluation of multimodal functions (F8–F23), EMRA outperforms the other algorithms in two out of the six functions. In terms of averages, as shown in table 8, EMRA achieves an average of (-9.10E+03) in F8, ranking 10th, while CSA, HHO and WOA perform significantly better. In F9, EMRA records (0.00E+00), ranking first, matching the results of CSA, HHO, and CGO. In F10, EMRA ranks 2nd with (4.441E-16), only slightly behind CSA (4.44E-16) and HHO, which share the same value. For F11, EMRA ranks first with an average of (0.00E+00), sharing the top spot with CSA, HHO, WOA, GWO and CGO. In F12, EMRA ranks 3rd with an average of (3.83E-10), behind CSA and CGO. In F13, EMRA achieves (7.38E-09), ranking 2nd, outperforming HHO (1.07E-06) among the top three ranking.

Table 8: Statistical comparison of EMRA and other metaheuristic algorithms in multimodal standard benchmark functions (F8–F13).

Algorithm	Function/ Metric	F8	F9	F10	F11	F12	F13
EMRA	Average.	-9.10E+03	0	4.44E-16	0	3.83E-10	7.38E-09
	Std.	2.17E+03	0	3.01E-31	0	7.34E-10	1.56E-08
	Ranking	10	1	1	1	3	2
CSA	Average.	-1.26E+04	0	4.44E-16	0	1.57E-32	1.35E-32
	Std.	1.85E-12	0	3.01E-31	0	1.11E-47	5.57E-48
	p-value	3.36E-12	−	3.21E-01	−	5.92E-03	1.21E-02
	Ranking	1	1	1	1	1	1
SCA	Average.	-1.15E+04	3.20E+01	1.25E-01	1.60E-06	1.01E+00	2.24E+00
	Std.	1.62E+03	4.88E+01	6.54E-01	8.56E-06	1.68E+00	7.87E-01
	p-value	1.27E-05	6.73E-04	3.01E-01	3.11E-01	1.77E-03	1.86E-22
	Ranking	9	9	7	7	10	10
SSA	Average.	-1.21E+04	8.95E+00	8.07E-01	1.50E-02	8.85E-02	5.28E-02
	Std.	1.23E+03	1.39E+01	1.00E+00	1.17E-02	3.19E-01	8.18E-02
	p-value	1.50E-08	8.33E-04	4.58E-05	2.57E-09	1.35E-01	8.04E-04
	Ranking	7	7	9	9	7	6
HHO	Average.	-1.26E+04	0	4.44E-16	0	8.83E-08	1.07E-06
	Std.	4.62E-05	0	3.01E-31	0	3.96E-07	4.11E-06
	p-value	3.36E-12	−	1.00E+00	−	2.29E-01	1.63E-01
	Ranking	2	1	1	1	4	3
WOA	Average.	-1.25E+04	1.89E-15	4.71E-15	0	1.20E-04	1.50E-02
	Std.	6.49E+02	1.04E-14	2.86E-15	0	5.70E-04	5.92E-02
	p-value	4.14E-11	3.21E-01	3.26E-11	−	2.55E-01	1.70E-01
	Ranking	4	5	5	1	5	5
PSO	Average.	-1.22E+04	2.00E+01	1.24E+00	5.92E-03	3.99E-01	2.81E-01
	Std.	1.09E+03	3.12E+01	1.52E+00	8.57E-03	9.57E-01	1.47E+00
	p-value	3.96E-09	8.90E-04	3.78E-05	3.72E-04	2.61E-02	3.00E-01
	Ranking	6	8	10	8	8	7
TSA	Average.	-1.23E+04	1.66E-03	4.96E-03	2.16E-02	7.17E-03	3.09E-01
	Std.	6.98E+02	8.86E-03	2.10E-02	6.59E-02	8.56E-03	3.90E-01
	p-value	1.45E-10	3.10E-01	2.02E-01	7.75E-02	2.40E-05	5.90E-05
	Ranking	5	6	6	10	6	8
GWO	Average.	-1.15E+04	5.24E+01	2.39E-01	0	4.01E-01	3.57E-01
	Std.	1.64E+03	4.56E+01	9.10E-01	0	1.15E+00	2.82E-01
	p-value	8.64E-06	4.56E-08	1.55E-01	−	6.21E-02	4.00E-09
	Ranking	8	10	8	1	9	9
CGO	Average.	-1.25E+04	0	1.98E-15	0	1.61E-21	1.05E-02
	Std.	6.49E+02	0	1.79E-15	0	3.78E-21	1.75E-02
	p-value	4.13E-11	−	1.59E-05	−	5.92E-03	1.73E-03
	Ranking	3	1	4	1	2	4

Regarding consistency, EMRA shows zero standard deviation in F9 and F11, confirming absolute stability, similar to CSA, HHO, and CGO. However, in F8, EMRA has a high standard deviation of (2.17E+03), whereas CSA (1.85E-12) and HHO (4.62E-05) display greater stability. In F10, EMRA achieves (3.009E-31), maintaining strong consistency close to CSA (3.01E-31). In F12, EMRA's standard deviation is (7.34E-10), higher than CSA (1.11E-47) but lower than SCA (1.68E+00), suggesting better stability compared to weaker algorithms. In F13, EMRA has a standard deviation of (1.56E-08), ranking among the more stable algorithms but slightly higher than CSA.

The p-value analysis highlights EMRA's statistical differences across F8 to F13. In F8, a p-value of (3.36E-12) confirms that CSA and HHO outperform EMRA. In F9, the p-value of equals sign indicates no difference, as multiple algorithms achieve the same result. F10 (1.59E-05) and F11 (3.11E-01) confirm

EMRA's advantage over weaker algorithms, while F12 (1.77E-03) and F13 (8.04E-04) highlight its significant superiority over competitors like SCA and SSA. Overall, EMRA exhibits statistically significant advantages in F10, F11, F12, and F13, confirming its strong performance in these functions. However, the high p-value in F9 indicates similar performance across multiple algorithms, while the low p-value in F8 suggests that CSA and HHO maintain a distinct advantage over EMRA in that function.

The performance of various algorithms was assessed using multimodal benchmark functions, with the rankings reflecting their effectiveness in consistently finding near-optimal solutions. CSA exhibited excellent performance, ranking first across multiple functions (F8, F9, F10, F11, F12, and F13). EMRA also performed strongly, ranking first for functions F9, F10, and F11, and securing top-tier positions in others. Conversely, algorithms such as SCA, SSA, and PSO frequently ranked lower, highlighting their relatively weaker effectiveness on these multimodal problems. These findings clearly underline the robust capability of algorithms like CSA and EMRA in navigating complex, multimodal search spaces.

4.6.3. Fixed-Dimension Multimodal Standard Benchmark Functions (F14–F23)

As shown in tables 9 and 10, EMRA outperforms several competing algorithms in F14, F18 and F20. For instance, in F14, EMRA achieves an average of (1.39E+00) competing closely with SSA and PSO, while in F18, it reaches (1.28E+01), outperforming weaker algorithms but still trailing behind SCA. In F20, EMRA records an average of (-2.69E+00), competing with CSA and SCA, which achieves a significantly better result. However, in F21, F22, and F23, EMRA lags behind, with averages of (-7.60E+00), (-7.21E+00), and (-7.29E+00), achieving an average performance higher than CSA and other top performers. While EMRA demonstrates promising performance in some functions, it struggles in others, indicating room for further optimization and enhancement.

In terms of consistency, EMRA exhibits stable performance in F15 and F17, with relatively low standard deviations of (2.88E-04) and (2.63E-02), demonstrating its reliability in these functions. However, in F14, F19 and F20, EMRA shows higher variability, with standard deviations reaching (2.13E+00), (6.11E-02), and (1.79E-01), respectively, indicating fluctuations in performance. In comparison, CSA and other top-ranking algorithms maintain near-zero standard deviations in these cases, reinforcing their superior stability over EMRA.

EMRA achieves statistically significant differences in F15 and F18, where p-values of (1.57E-03) and (1.14E-07) confirm that it performs competitively in these functions. However, in F14, F16, F21, and F22, p-values below (1.00E-08) indicate that CSA and other leading algorithms significantly outperform EMRA. In F19 and F20, EMRA's performance is statistically weaker than that of CSA and SSA, reinforcing the need for improvement in these functions. Overall, EMRA performs well in F15 and F18, showing competitive averages and stable results, but struggles in most other fixed-dimension multimodal functions, particularly in F21 to F23. The high standard deviations and low p-values in several cases indicate that EMRA is less consistent compared to top-ranking algorithms like CSA and SSA. While it demonstrates strengths in select functions, further improvements are needed to enhance its robustness across all fixed-dimension multimodal benchmarks.

Analyzing the ranking of EMRA in multimodal fixed-dimension benchmark functions (F14–F23) demonstrates competitiveness in specific functions, particularly attaining slightly higher rankings in F14 which achieved 7th place, and F18 and F20, both of which achieved 8th place. However, despite these positive outcomes, EMRA generally ranked lower across this benchmark set, occupying ranks of 9th and 10th positions in multiple functions, such as F15, F16, F17, F19, F21, F22, and F23. These consistently lower rankings indicate that EMRA faces challenges in reliably achieving top tier positions across multimodal fixed-dimension problems, highlighting opportunities for further enhancement in algorithm robustness and performance stability.

Table 9: Statistical comparison of EMRA and other metaheuristic algorithms in fixed-dimension multimodal standard benchmark functions (F14–F18).

Algorithm	Function/ Metric	F14	F15	F16	F17	F18
EMRA	Average.	1.39E+00	7.69E-04	-9.65E-01	4.30E-01	1.28E+01
	Std.	2.13E+00	2.88E-04	5.77E-02	2.63E-02	8.84E+00
	Ranking	7	9	9	9	8
CSA	Average.	9.98E-01	1.67E-03	3.69E-222	8.45E-01	3.27E+01
	Std.	3.39E-16	1.10E-18	0.00E+00	9.16E-16	1.45E-14
	p-value	3.21E-01	1.82E-24	1.96E-64	5.51E-63	7.37E-18
	Ranking	1	10	10	10	10
SCA	Average.	2.03E+00	6.48E-04	-1.03E+00	4.00E-01	3.00E+00
	Std.	1.85E+00	4.88E-04	1.17E-04	2.13E-03	4.48E-05
	p-value	2.15E-01	2.46E-01	4.46E-08	5.80E-08	1.14E-07
	Ranking	9	8	8	8	6
SSA	Average.	1.06E+00	5.45E-04	-1.03E+00	3.98E-01	3.00E+00
	Std.	2.52E-01	2.31E-04	1.58E-14	7.70E-15	1.87E-13
	p-value	4.13E-01	1.57E-03	4.29E-08	1.02E-08	1.14E-07
	Ranking	6	7	3	3	3
HHO	Average.	9.98E-01	3.30E-04	-1.03E+00	3.98E-01	3.00E+00
	Std.	1.15E-11	2.38E-05	4.91E-10	6.74E-06	1.43E-06
	p-value	3.21E-01	1.73E-11	4.29E-08	1.02E-08	1.14E-07
	Ranking	3	2	4	4	4
WOA	Average.	1.39E+00	3.13E-04	-1.03E+00	3.98E-01	3.90E+00
	Std.	1.79E+00	9.85E-06	5.83E-10	1.51E-05	4.94E+00
	p-value	9.96E-01	4.78E-12	4.29E-08	1.03E-08	1.18E-05
	Ranking	8	1	5	6	7
PSO	Average.	1.03E+00	3.91E-04	-1.03E+00	3.98E-01	3.00E+00
	Std.	1.81E-01	1.07E-04	0.00E+00	1.13E-16	3.03E-15
	p-value	3.66E-01	7.72E-09	4.29E-08	1.02E-08	1.14E-07
	Ranking	5	5	1	1	2
TSA	Average.	9.98E-01	4.07E-04	-1.03E+00	3.98E-01	3.00E+01
	Std.	1.53E-06	3.50E-04	5.86E-06	1.17E-04	3.68E+00
	p-value	3.21E-01	5.19E-05	4.29E-08	1.06E-08	5.18E-14
	Ranking	4	6	7	7	9
GWO	Average.	2.89E+00	3.68E-04	-1.03E+00	3.98E-01	3.00E+00
	Std.	3.64E+00	9.97E-05	8.09E-08	3.89E-06	2.50E-05
	p-value	5.56E-02	1.31E-09	4.29E-08	1.02E-08	1.14E-07
	Ranking	10	4	6	5	5
CGO	Average.	9.98E-01	3.38E-04	-1.03E+00	3.98E-01	3.00E+00
	Std.	3.39E-16	1.67E-04	0	1.13E-16	4.52E-16
	p-value	3.21E-01	2.06E-09	4.29E-08	1.02E-08	1.14E-07
	Ranking	1	3	1	1	1

Table 10: Statistical comparison of EMRA and other metaheuristic algorithms in fixed-dimension multimodal standard benchmark functions (F19–F23).

Algorithm	Function/ Metric	F19	F20	F21	F22	F23
EMRA	Average.	-3.74E+00	-2.69E+00	-7.60E+00	-7.21E+00	-7.29E+00
	Std.	6.11E-02	1.79E-01	2.59E+00	2.65E+00	2.69E+00
	Ranking	9	8	9	10	10

Table 10: continue

CSA	Average.	-1.90E+00	-1.17E+00	-1.02E+01	-1.04E+01	-1.05E+01
	Std.	6.78E-16	2.26E-16	3.61E-15	0.00E+00	3.61E-15
	p-value	3.81E-79	1.41E-47	1.37E-06	1.40E-08	1.39E-08
	Ranking	10	10	1	1	2
SCA	Average.	-3.85E+00	-2.64E+00	-6.75E+00	-7.73E+00	-8.49E+00
	Std.	4.09E-03	3.96E-01	2.12E+00	2.22E+00	2.06E+00
	p-value	1.04E-14	5.58E-01	1.68E-01	4.20E-01	5.81E-02
	Ranking	7	9	10	9	9
SSA	Average.	-3.86E+00	-3.22E+00	-9.65E+00	-9.88E+00	-1.00E+01
	Std.	5.58E-11	5.72E-02	1.54E+00	1.61E+00	1.64E+00
	p-value	1.84E-16	1.88E-22	4.64E-04	1.62E-05	1.64E-05
	Ranking	3	6	6	6	6
HHO	Average.	-3.86E+00	-3.09E+00	-9.98E+00	-1.00E+01	-1.04E+01
	Std.	2.31E-03	1.07E-01	9.31E-01	1.35E+00	9.87E-01
	p-value	3.82E-16	3.28E-15	1.47E-05	2.48E-06	2.41E-07
	Ranking	5	7	2	5	4
WOA	Average.	-3.86E+00	-3.31E+00	-9.81E+00	-1.04E+01	-1.05E+01
	Std.	7.95E-03	4.29E-02	1.29E+00	3.31E-03	9.49E-04
	p-value	1.96E-15	4.41E-26	1.01E-04	1.41E-08	1.40E-08
	Ranking	6	1	5	2	3
PSO	Average.	-3.86E+00	-3.26E+00	-8.97E+00	-8.91E+00	-1.00E+01
	Std.	2.71E-15	8.32E-02	2.18E+00	2.54E+00	1.65E+00
	p-value	1.84E-16	9.91E-23	3.10E-02	1.39E-02	1.68E-05
	Ranking	1	5	7	7	7
TSA	Average.	-3.81E+00	-3.28E+00	-9.97E+00	-1.04E+01	-1.03E+01
	Std.	1.95E-01	8.05E-02	9.22E-01	2.57E-02	9.79E-01
	p-value	6.05E-02	8.35E-24	1.58E-05	1.61E-08	2.56E-07
	Ranking	8	4	3	3	5
GWO	Average.	-3.86E+00	-3.30E+00	-8.58E+00	-8.91E+00	-9.39E+00
	Std.	2.82E-03	5.47E-02	2.27E+00	2.07E+00	1.88E+00
	p-value	3.54E-16	1.82E-25	1.26E-01	7.83E-03	9.07E-04
	Ranking	4	2	8	8	8
CGO	Average.	-3.86E+00	-3.29E+00	-9.82E+00	-1.02E+01	-1.05E+01
	Std.	2.71E-15	5.11E-02	1.28E+00	9.63E-01	9.03E-15
	p-value	1.84E-16	2.60E-25	9.62E-05	2.34E-07	1.39E-08
	Ranking	1	3	4	4	1

As shown in table 11, our proposed EMRA algorithm achieved a respectable mean rank of 5.5, demonstrating strong capabilities across various problem types. CGO obtained the best overall performance with the lowest score of 2.5, followed by HHO and CSA with scores of 3.3 and 3.7, respectively. WOA also performed well, earning a rank of 4.6. In contrast, SCA recorded the highest mean rank of 8.2, indicating comparatively weaker results. While EMRA performed notably well in several benchmarks, particularly those involving unimodal and multimodal problems, CGO and HHO delivered more consistent and superior performance across the entire test suite.

Table 11: Ranking score comparison of EMRA, CSA, SCA, SSA, HHO, WOA, PSO, TSA, GWO and CGO by standard benchmark functions.

Algorithm	CGO	HHO	CSA	WOA	EMRA	PSO	SSA	TSA	GWO	SCA
Ranking Score	2.5	3.3	3.7	4.6	5.5	6.3	6.4	6.5	6.7	8.2

4.7. Evaluating the EMRA in Comparison with others Using CEC2019 Benchmark Functions

As shown in tables 12 and 13, EMRA demonstrates outstanding performance in F1, where it achieves an optimal solution with an average of (1.00E+00), securing the top rank among all algorithms. Additionally, EMRA maintains perfect stability in F1 with a zero-standard deviation, proving its consistency in this function. EMRA also shows competitive results in F3 and F6, achieving averages of 1.37E+01 and 1.34E+01, respectively. Despite ranking lower than the top-performing algorithms, its performance in these functions remains consistent with relatively low standard deviations (5.40E-04) in F3 and (7.88E-01) in F6. The p-values in these cases indicate that EMRA is statistically comparable to some of the best-performing algorithms. In F8, EMRA achieves an average of 7.33E+00, demonstrating reliable performance in these functions. Its standard deviation (3.21E-01) indicates a balanced level of stability compared to several competing algorithms.

Regarding consistency, EMRA demonstrates strong consistency in multiple functions, particularly F1, F7, and F10, where its standard deviations remain within an acceptable range compared to most competing algorithms. EMRA maintains perfect stability in F1, with a zero-standard deviation, proving its consistency in this function. In F7, EMRA records a standard deviation of (1.55E+02) which, while not the lowest, still allows it to maintain a stable consistency. In F10, EMRA achieves a standard deviation of (1.09E-01), indicating balanced performance. Additionally, while EMRA does not rank at the top in F2 and F4, its standard deviations of (3.65E+03) and (5.10E+03) are comparable to some competing methods, demonstrating its ability to remain within an expected performance range.

The p-value analysis identified the statistical significance of EMRA between F1 and F10. Specifically, in F1 with a p-value of 0.00E+00, it validates that EMRA reaches the best solution and is among the best-performing algorithms with no statistical difference. Furthermore, in F3 and F6, the p-values indicate that EMRA is competitively performing since it reflects consistent results with slight differences from the best-performing algorithms. For F8, the (1.07E-12) p-value illustrates a significant statistical distinction, reflecting EMRA's high ranking in this function. Similarly, for F10, a p-value of (2.18E-08) reflects that EMRA's performance is stable and statistically comparable to numerous superior algorithms. While there are functions with lower rankings, such as F5 and F9, their p-values reflect that there is room for improvement, i.e., EMRA is able to enhance its performance in these functions. In conclusion, the p-value analysis validates that EMRA is statistically significant related to the key benchmark functions, enhancing its consistency and reliability in the CEC-2019 tests.

Table 12: Statistical comparison of EMRA and other metaheuristic algorithms by CEC2019 benchmark functions (F1-F5).

Algorithm	Function/ Metric	F1	F2	F3	F4	F5
EMRA	Average.	1.00E+00	1.60E+04	1.37E+01	1.66E+04	6.46E+00
	Std.	0.00E+00	3.65E+03	5.40E-04	5.10E+03	8.00E-01
	Ranking	1	10	9	9	9
CSA	Average.	6.48E+05	1.95E+01	1.37E+01	4.41E+04	9.13E+00
	Std.	6.51E-10	3.61E-15	9.03E-15	2.22E-11	1.81E-15
	p-value	0.00E+00	1.02E-31	5.77E-62	1.10E-36	8.55E-26
	Ranking	5	9	10	10	10
SCA	Average.	5.06E+04	1.86E+01	1.37E+01	3.90E+03	3.56E+00
	Std.	5.03E+03	9.55E-02	1.82E-04	1.14E+03	1.28E-01
	p-value	8.58E-52	1.02E-31	1.04E-06	3.12E-19	3.03E-27
	Ranking	3	8	8	8	7

Table 12: continue

SSA	Average.	1.06E+09	1.83E+01	1.37E+01	3.67E+01	2.25E+00
	Std.	2.17E+09	2.62E-02	4.57E-04	2.14E+01	1.35E-01
	p-value	9.90E-03	1.02E-31	3.96E-05	4.01E-25	1.12E-35
	Ranking	10	4	7	2	3
HHO	Average.	5.23E+04	1.84E+01	1.37E+01	2.20E+02	3.59E+00
	Std.	5.07E+03	1.14E-02	6.61E-06	7.14E+01	6.59E-01
	p-value	2.18E-52	1.02E-31	1.50E-08	6.95E-25	8.38E-22
	Ranking	4	6	6	5	8
WOA	Average.	7.49E+05	1.83E+01	1.37E+01	4.03E+02	2.86E+00
	Std.	1.89E+06	5.62E-03	8.32E-07	4.76E+02	4.56E-01
	p-value	3.39E-02	1.02E-31	1.05E-08	1.63E-24	3.44E-29
	Ranking	6	5	4	6	5
PSO	Average.	7.81E+08	1.83E+01	1.37E+01	2.39E+01	2.14E+00
	Std.	1.42E+09	3.61E-15	9.03E-15	1.32E+01	9.18E-02
	p-value	3.80E-03	1.02E-31	1.03E-08	3.86E-25	1.70E-36
	Ranking	8	1	1	1	2
TSA	Average.	9.79E+08	1.85E+01	1.37E+01	1.21E+03	3.15E+00
	Std.	3.38E+09	1.30E-01	9.49E-07	1.16E+03	5.96E-01
	p-value	1.18E-01	1.02E-31	1.06E-08	4.78E-23	1.44E-25
	Ranking	9	7	5	7	6
GWO	Average.	2.74E+07	1.83E+01	1.37E+01	1.43E+02	2.55E+00
	Std.	6.64E+07	5.42E-04	5.36E-07	4.42E+02	2.69E-01
	p-value	2.78E-02	1.02E-31	1.04E-08	6.60E-25	4.96E-33
	Ranking	7	3	3	4	4
CGO	Average.	3.38E+04	1.83E+01	1.37E+01	3.86E+01	2.13E+00
	Std.	1.70E+03	3.61E-15	9.03E-15	2.21E+01	8.37E-02
	p-value	8.67E-69	1.02E-31	1.03E-08	4.03E-25	1.51E-36
	Ranking	2	1	1	3	1

Table 13: Statistical comparison of EMRA and other metaheuristic algorithms by CEC2019 benchmark functions (F6-F10).

Algorithm	**Function/** Metric	**F6**	**F7**	**F8**	**F9**	**F10**
EMRA	Average.	1.34E+01	1.20E+03	7.33E+00	2.50E+03	2.18E+01
	Std.	7.88E-01	1.55E+02	3.21E-01	7.04E+02	1.09E-01
	Ranking	10	9	9	9	9
CSA	Average.	1.27E+01	2.16E+03	7.99E+00	4.71E+03	2.19E+01
	Std.	8.61E-01	1.66E+02	2.36E-01	1.85E-12	3.88E-02
	p-value	2.09E-03	6.45E-31	1.07E-12	1.76E-24	2.18E-08
	Ranking	9	10	10	10	10
SCA	Average.	1.22E+01	8.61E+02	5.99E+00	3.11E+02	2.14E+01
	Std.	5.90E-01	1.89E+02	4.24E-01	1.21E+02	2.64E-01
	p-value	3.82E-09	2.76E-10	6.53E-20	6.83E-24	2.13E-08
	Ranking	8	8	5	8	7
SSA	Average.	6.10E+00	3.53E+02	5.49E+00	3.61E+00	2.10E+01
	Std.	1.74E+00	2.07E+02	7.35E-01	1.77E-01	1.03E-01
	p-value	8.59E-29	2.22E-25	3.68E-18	5.02E-27	1.26E-34
	Ranking	1	3	4	3	2
HHO	Average.	1.07E+01	5.30E+02	6.16E+00	4.30E+00	2.13E+01
	Std.	1.01E+00	1.71E+02	3.47E-01	4.06E-01	1.27E-01
	p-value	6.22E-17	9.01E-23	1.60E-19	5.09E-27	2.57E-23
	Ranking	3	5	7	4	4
WOA	Average.	1.06E+01	6.16E+02	6.10E+00	6.23E+00	2.13E+01
	Std.	1.26E+00	2.95E+02	6.28E-01	2.03E+00	1.63E-01
	p-value	7.34E-15	1.34E-13	1.65E-13	5.29E-27	5.50E-19
	Ranking	2	7	6	5	5

Table 13: continue

PSO	Average.	1.10E+01	1.63E+02	4.68E+00	3.39E+00	2.01E+01
	Std.	1.48E+00	1.60E+02	1.18E+00	2.76E-02	4.87E+00
	p-value	7.29E-11	3.10E-33	4.16E-17	4.99E-27	5.99E-02
	Ranking	5	1	2	2	1
TSA	Average.	1.10E+01	5.78E+02	6.28E+00	1.19E+01	2.14E+01
	Std.	1.16E+00	2.26E+02	6.17E-01	1.17E+01	1.44E-01
	p-value	1.16E-13	5.26E-18	2.75E-11	5.98E-27	5.16E-16
	Ranking	4	6	8	6	6
GWO	Average.	1.19E+01	4.61E+02	4.82E+00	1.31E+01	2.15E+01
	Std.	9.01E-01	2.53E+02	1.00E+00	4.15E+01	1.29E-01
	p-value	2.72E-09	9.64E-20	7.16E-19	6.63E-27	4.84E-13
	Ranking	7	4	3	7	8
CGO	Average.	1.11E+01	2.36E+02	3.75E+00	3.34E+00	2.11E+01
	Std.	7.00E-01	2.58E+02	1.43E+00	3.20E-03	9.37E-02
	p-value	3.88E-17	7.27E-25	2.04E-19	4.99E-27	1.07E-33
	Ranking	6	2	1	1	3

Using the Friedman mean ranking score on the 10 CEC2019 benchmark functions, as presented in table 14, EMRA scored 8.4, indicating moderate performance. While it showed strength in specific cases, CGO 2.1 and PSO 2.4 achieved the top ranks, reflecting greater consistency. SSA also performed well with a score of 3.9, whereas CSA had the weakest performance with a score of 9.3. These results highlight that EMRA was competitive in some functions, especially Function 1 where it achieved the optimal value, but it was outperformed by the most robust algorithms across the suite.

Table 14: Comparison of the ranking scores of EMRA with different metaheuristics on CEC2019.

Algorithm	CGO	HHO	CSA	WOA	EMRA	PSO	SSA	TSA	GWO	SCA
Ranking Score	2.1	2.4	3.9	5	5.1	5.2	6.4	7	8.4	9.3

4.8. Applying EMRA to Deal with Engineering Design Challenges

This section mainly focuses on evaluating how well EMRA performs in real world problems. To do this, three classic engineering-design problems were used: (welded beam, pressure vessel, and tension spring designs). A brief theoretical overview of each issue is presented. EMRA's performance was evaluated against various existing algorithms, including MRA, CSA, HHO, WOA, and TSA. This comparison aims to illustrate EMRA's capability and competitiveness in handling complicated design optimization challenges. The methodology provides the average and standard deviations obtained after running the selected algorithm 30 times on the problems.

4.8.1. Welded Beam Design

In this problem, the main objective is to reduce the total construction cost of a welded-beam structure that is subject to a variety of physical and manufacturing constraints, as shown in the welded beam design diagram in figure 3. The welded thickness(h), beam length (l), depth (b), and height of bar (t) are the decision variables. Shear stress, bending stress, deflection, and construction feasibility are the criteria used to establish the constraints.

Assume: $\vec{w} = [w_1\ w_2\ w_3\ w_4] = [h\ l\ t\ b]$

Objective function: $f_{\vec{w}} = 1.10471 w_1^2 w_2 + 0.04811 w_3 w_4 (14.0 + w_2)$ (11)

The design must satisfy the following constraints:

$g_1(w) = \tau(w) - 13{,}600 \leq 0$

$g_2(w) = \sigma(w) - 30{,}000 \leq 0$

$g_3(\vec{w}) = \delta(w) - 0.25 \leq 0$

$g_4(w) = w - w_4 \leq 0$

$g_5(\vec{w}) = 600 - w_4 \leq 0$

The variable bounds are given as: $0.1 \leq w_1 \leq 2,\ 0.1 \leq w_2 \leq 10,\ 0.1 \leq w_3 \leq 10,\ 0.1 \leq w_4 \leq 2$

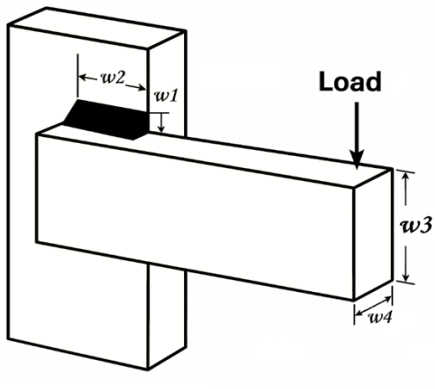

Figure 3: Welded Beam Design.

4.8.2. Pressure Vessel Design

The pressure vessel design's challenge to optimize the manufacturing cost of a cylindrical pressure vessel. The cost function considers the material, forming, and welding expenses. The design parameters are the shell thickness (Ts), head thickness (Th), inner radius (R), and length of the cylindrical part (L) excluding the head itself. Figure 4 shows the problem variables that are suitable for optimization. The problem variables, equation, and constraints are displayed below.

Assume: $\vec{p} = [p_1\ p_2\ p_3\ p_4] = [T_s\ T_h\ R\ L]$

Objective function: $f_{\vec{p}} = 0.6224p_1p_3\ p_4 + 1.778p_2p_3^2 + 3.1661p_1^2p_4 + 19.84p_1^2p_3$ (12)

The design must satisfy the following constraints:

$g_1(\vec{p}) = -p_1 + 0.0193p_3 \leq 0$

$g_2(\vec{p}) = -p_3 + 0.0095p_3 \leq 0$

$g_3(\vec{p}) = -\pi p_3^2 p_4 - \frac{4}{3}\pi p_3^3 + 1{,}296{,}000 \leq 0$

$g_4(\vec{p}) = p_4 - 240 \leq 0$

The variable bounds are given as: $0 \leq p_1 \leq 99,\ 0 \leq p_2 \leq 99,\ 10 \leq p_3 \leq 200,\ 10 \leq p_4 \leq 200$

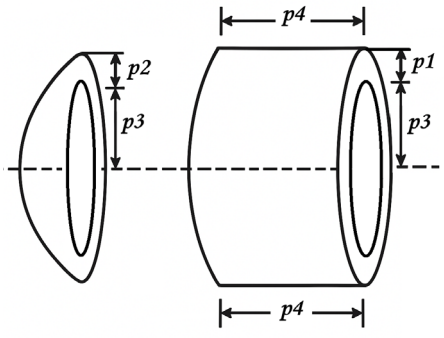

Figure 4: Pressure vessel design.

4.8.3. Tension Spring Design

The goal of the present application is to reduce the size or weight of a compression spring while continuing to meet the mechanical and geometric constraints on it, such as the limits for shear stress, deflection, and surge frequency. Figure 5 highlights the variables of the problem that are suitable for optimization.

Assume: $\vec{t} = [t_1\ t_2\ t_3\ ] = [d\ D\ L]$

$$\text{Objective function: } f_{(t)} = (2 + t_3)t_1^2 t_2 \qquad (13)$$

The design must satisfy the following constraints:

$$g_1(\vec{t}) = 1 - \frac{t_2^3 t_3}{71785 t_1^4} \leq 0$$

$$g_2(\vec{t}) = \frac{4t_2^2 - t_1 t_2}{71785 t_1^4} + \frac{1}{5108 t_1^2} \leq 0$$

$$g_3(\vec{t}) = 1 - \frac{140.45 t_1}{t_2^2 t_3} \leq 0$$

$$g_4(\vec{t}) = \frac{t_1 + t_2}{1.5} - 1 \leq 0$$

The variable bounds are given as: $0.05 \leq t_1 \leq 2.00$, $0.25 \leq t_2 \leq 1.30$, $2.00 \leq t_3 \leq 15.00$.

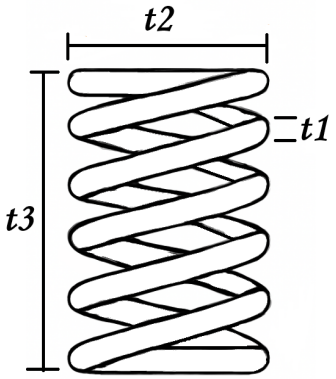

Figure 5: Tension Spring Design

4.8.4. Comparative Analysis of EMRA and MRA Through Three Engineering Applications

Comparative analysis was conducted to evaluate the performance of the proposed EMRA against the original version, MRA, using three established engineering design problems: pressure vessel design, welded beam design, and tension / compression spring design. The average performance data presented in table 15 demonstrates that EMRA generally exhibits more stable performance compared to the original MRA across the three evaluated engineering design problems. In the welded beam design, EMRA attains a lower average cost of (1.89E+00), in contrast to MRA's (2.05E+00), indicating a more efficient design with lower construction costs. In the tension spring design, EMRA obtains a slightly better average value of (1.32E-02), marginally lower than MRA's (1.33E-02), indicating improved optimization accuracy. However, in the pressure vessel design, the outcomes indicate an agreement. MRA achieves a better average cost of (7.77E+03) in comparison with EMRA's (9.81E+03). In terms of standard deviations, EMRA performs more consistently in two of the three cases. For the welded beam design, EMRA shows a notably smaller standard deviation of (2.34E-01) compared to MRA's (5.31E-01), indicating more consistent outcomes across multiple runs. In the tension spring design, EMRA also maintains a lower standard deviation of (9.74E-05), while MRA records a significantly higher value of (5.66E-04), confirming EMRA's superior repeatability. Conversely, in the pressure vessel design, MRA demonstrates greater stability, with a lower standard deviation of (9.51E+02), compared to EMRA's (2.11E+03). Overall, EMRA delivers better or comparable average solutions in two of the three cases and consistently yields lower standard deviations in most applications. These findings suggest that EMRA is a more reliable and robust optimization algorithm when applied to constrained engineering design problems.

Table 15: Performance of EMRA against the original MRA in three engineering applications.

Engineering App. Design	EMRA		MRA	
	Average	Std.	Average	Std.
Welded Beam	1.89E+00	2.34E-01	2.05E+00	5.31E-01
Pressure Vessel	9.81E+03	2.11E+03	7.77E+03	9.51E+02
Tension Spring	1.32E-02	9.74E-05	1.33E-02	5.66E-04

4.8.5. Comparative Analysis of EMRA, CSA, HHO, WOA, and TSA in Engineering Applications

A comparison analysis was conducted to evaluate the performance of the proposed EMRA compared to four recognized metaheuristic algorithms, CSA, HHO, WOA, and TSA, which were selected for their demonstrated productivity in addressing complex optimization challenges. The evaluation was performed using three established engineering design requirements to assess EMRA's efficiency, robustness, consistency, and the ranking was recognized based on the average performance of each algorithm. Additionally, it provided a comparative measure of their effectiveness in solving the optimization tasks.

The outcomes are shown in table 16. In terms of average performance, EMRA performs very competitively across all three engineering design problems. In the welded beam design, EMRA achieves an average cost of (1.89E+00), which ranks first among the algorithms. In the pressure vessel design, EMRA has an average value of (9.81E+03), which places it in fourth place, behind CGO, GWO, and SCA. However, EMRA excels in the tension spring design, with an average of (1.32E-02), securing the first position, outperforming all other algorithms.

In terms of standard deviations, which reflect the consistency of the optimization results, EMRA exhibits strong stability across the three engineering problems. For the welded beam design, EMRA has a standard deviation of (2.34E-01), the lowest among the algorithms, indicating the most consistent performance. In the pressure vessel design, EMRA again shows competitive consistency with a standard deviation of (2.11E+03), which is better than many of the algorithms, including CSA (4.06E+03) and WOA (2.85E+03). For the tension spring design, EMRA records an impressive standard deviation of (9.74E-05) which is also the lowest, showcasing its highly reliable convergence compared to other algorithms like CSA (8.71E+07) and TSA (1.36E-03). Overall, EMRA exhibits competitive performance in every scenario, providing great performance and exceptional consistency, particularly in the tension spring design where it outperforms in terms of stability and average performance.

Table 16: Performance of EMRA with CSA, HHO, WOA and TSA in three engineering applications.

Algorithm	Engineering Applications Design								
	Welded Beam			Pressure Vessel			Tension Spring		
	Avg	Std.	Rank	Avg.	Std.	Rank	Avg.	Std.	Rank
EMRA	1.89E+00	2.34E-01	1	9.81E+03	2.11E+03	4	1.32E-02	9.74E-05	1
CSA	1.92E+00	4.42E-01	3	7.66E+03	4.06E+03	2	1.62E+07	8.71E+07	5
HHO	1.90E+00	4.01E-01	2	6.72E+03	4.37E+02	1	1.40E-02	1.20E-03	3
WOA	2.54E+00	8.47E-01	4	9.58E+03	2.85E+03	3	1.38E-02	1.16E-03	2
TSA	1.17E+07	6.02E+07	5	1.58E+04	8.42E+03	5	1.40E-02	1.36E-03	4

5. Discussion

This section analyzes the efficiency and stability of the proposed EMRA by examining its performance over 23 standard benchmark functions, as well as 10 CEC2019 competition benchmark functions. These benchmark functions are widely used by researchers to assess the performance of newly created algorithms in a standard manner across the variety of methods [2]. These functions are also used as benchmarks to evaluate how optimization algorithms perform across different problem types. They include complex features like rotation, scaling, and shifting to better simulate real-world optimization challenges [41].

For these three sets of standard benchmark functions, unimodal functions with one global optimal solution and those without a local optimal solution were determined to be suitable for the evaluation

of an algorithm's exploiting ability. Multimodal functions are a popular choice for exploration-oriented optimization problems as they provide a useful benchmark for evaluating an algorithm's capacity for dealing with a landscape with many peaks and valleys to reach the global optimum. Fixed-dimension multimodal functions with multiple local solutions play a key role in evaluating an algorithm's ability of escaping from local optima to optimally seek the global optimum [42]. They have been widely utilized to benchmark the performance and robustness of optimization algorithms, which is beneficial for comparisons between EMRA and other algorithms [43]. To rank the performance of the algorithms across the 23 standard benchmark functions, we applied the Friedman Mean ranking method [44], which evaluates overall performance consistency where lower scores reflect better outcomes.

The CEC2019 benchmark functions are widely used to test optimization algorithms because they're complex and mimic real-world challenges. These benchmarks cover different types of functions (unimodal, multimodal, hybrid, and composition), helping to assess an algorithm's ability to both explore new solutions and make the most of what it already knows, striking the right balance between the two. Overall, they offer a fair way to compare algorithms and reflect real-world optimization problems [45]. Also, to systematically evaluate the effectiveness of different algorithms, we utilized ranking metrics based on its performance across multiple benchmark functions using the Friedman Mean Rank method [44]. The rankings reflect each algorithm's ability to consistently approach optimal solutions.

5.1. EMRA vs. MRA

The results from testing EMRA against MRA in the 23 standard benchmark functions indicate that EMRA significantly enhances the algorithm's ability to balance exploration and exploitation. EMRA is comparable to MRA in terms of unimodal functions; however, it surpasses it in terms of both average performance and consistency in some functions. It shows superior performance in multimodal and fixed-dimension multimodal functions, achieving better average fitness values and lower standard deviations. This demonstrates EMRA's improved capacity to avoid local optima through effective exploration, as well as more precise solution refinement through stronger exploitation. Nevertheless, some benchmark functions reveal areas where EMRA's performance could be further enhanced, suggesting the potential benefits of incorporating adaptive parameter strategies and diversity maintenance mechanisms. In summary, EMRA provides a more robust and reliable optimization framework compared to the original MRA, particularly for complex and high-dimensional problems.

The comparison shows that EMRA performs well in comparison with MRA in the CEC2019 benchmark functions, especially for complex optimization problems. EMRA overall achieved better performance compared to MRA in 6 of the 10 functions, especially in functions (F3, F4, F5, F7, F8, and F9), which demonstrated its ability to explore complex search spaces, escape local optimum solutions, and quickly converge on global minimum solutions. Moreover, EMRA resulted in improved stability for 8 functions and significantly lower standard deviations, demonstrating the robustness and efficiency of EMRA over multiple runs. Note that EMRA also has matched the best results solved by the original MRA in function F1, indicating that it does preserve some superiority from the original and further enhances its capability to handle more challenging optimization problems.

5.2. EMRA vs Other Metaheuristics

The results indicate that EMRA is competitive with other popular metaheuristic algorithms regarding the 23 standard benchmark functions and CEC2019 benchmark functions. In the 23-function test suite, EMRA performs well in both unimodal and multimodal benchmarks and takes the first place in F1–F4 and F7, with high stability and small standard deviations. Compared with CSA, SCA, SSA, EMRA is competitive with the best performing values in F9, F10, and F11, with stable and accurate results. For some functions like F8, EMRA is slightly dominated by algorithms such as CSA and HHO, which means that EMRA is also competitive in multimodal landscapes. EMRA shows better stability and accuracy in many functions, while it obtained a low rank for a few fixed-dimension multimodal functions (F14–F23), which implies that there is space for high complexity with stronger handling capability.

The performance of various algorithms was systematically ranked across 10 CEC 2019 benchmark functions (F1–F10). EMRA displayed notable strength for F1, achieving the highest-ranking position. However, EMRA demonstrated comparatively lower rankings in the remaining functions, mostly occupying ranks between 9th and 10th. Algorithms such as PSO and CGO consistently showed superior rankings across multiple functions, frequently securing the top positions. Conversely, algorithms like CSA generally held lower positions, indicating weaker overall effectiveness in the benchmark tests. These results illustrate EMRA's competitive capability in specific problems such as F1 but they also highlight areas where further refinement could improve its consistency across a wider variety of challenging optimization scenarios.

5.3. Engineering Applications in EMRA

The three benchmark problems of the welded beam, pressure vessel and tension spring design problems have been extensively used as the standard comparison benchmarks in the structural optimization literature. The welded beam problem [46] is a problem which contains multiple stresses, deflection and construction constraints. The pressure vessel problem is to minimize the cost of manufacture with stress and thickness constraints [3]. The tension spring problem, where the weight is to be minimized subject to constraints on shear stress, surge frequency and deflection. The spring is classified by three design parameters: wire diameter, mean coil diameter, and number of active coils [47]. These design problems illustrate practical scenarios in structural and mechanical engineering, each characterized by distinct objective functions, design variables, and sets of constraints [48], characterized by numerous constraints, and significant real-world importance [9].

Although EMRA delivers enhanced results compared to MRA across several engineering optimization problems, both algorithms still face difficulties in certain scenarios. For instance, in the pressure vessel design problem, MRA achieves a better average objective value, but EMRA exhibits greater variability, indicating variation in its search process. This reflects the complexity of handling non-linear constraints and multiple variables in such problems, where neither algorithm fully excels at all types of problem. Furthermore, while EMRA shows superior performance in the welded beam and tension spring problems, variations in solution quality designate the potential to improve the algorithm's convergence stability. These results highlight the requirement for the further fine-tuning of EMRA's parameters, potentially integrating hybrid methods to support its robustness and reliability when applied to diverse and challenging engineering design tasks.

The results demonstrate that EMRA is competitive compared to other meta-heuristic algorithms in engineering problems. It achieved the lowest average and highest stability, better than all compared algorithms, for the welded beam and tension spring problems. For the pressure vessel problem, EMRA ranked fourth yet demonstrated better consistency than many of the other methods. In general, EMRA showed powerful optimization ability and stability and was far superior to CSA, WOA, TSA, and even HHO in the solution of two out of the three real-world engineering design problems.

6. Conclusions

This study proposed the EMRA, an improved version of the original MRA, with a focus on better balancing the exploration and exploitation phases and preventing early convergence to local optima. EMRA was tested on 23 standard benchmark functions and 10 CEC2019 test functions, and its performance was compared with several well-known algorithms including (CSA, SCA, SSA, HHO, WOA, PSO, TSA, GWO, and CGO). The results showed that EMRA outperformed MRA in several cases, achieving optimal results in six standard functions, particularly in fixed-dimension multimodal benchmarks. It also performed better in six of the CEC2019 functions, and reached the optimum in one. Compared to other algorithms, EMRA showed consistent performance with lower standard deviations, indicating better stability and reliability. Additionally, EMRA was successfully applied to real-world engineering design problems such as (welded beam, pressure vessel, and tension spring design). In these tasks, EMRA ranked first in both the welded beam and tension spring problems, demonstrating its practical value. In conclusion, EMRA presents a promising and reliable optimization technique that

improves upon its predecessor and performs competitively with other advanced metaheuristic algorithms in both benchmark and real-world scenarios.

Although EMRA has shown promising improvements over the original MRA, there are still several areas worth exploring. One key direction is the development of an adaptive parameter control mechanism, especially for transition factor (K), to better balance exploration and exploitation based on the algorithm's progress. This could enhance convergence speed and solution quality. Another important goal is to reduce EMRA's computational cost. Improving its efficiency would allow the algorithm to scale to large, complex problems and support real-time applications where quick decisions are needed. Expanding the application of EMRA to the medical field such as in medical image processing, disease diagnosis, and treatment planning offers a promising opportunity to demonstrate its versatility and real-world value. Future research should also test EMRA on high-dimensional, nonlinear, and multi-objective problems in these domains. Lastly, improving the population diversity control mechanisms is essential to avoid premature convergence and ensure the broad exploration of the search space, leading to more robust and reliable solutions.

Author contribution: Shabaz Kawa Ali: Conceptualization, Data curation, Formal Analysis, Funding acquisition, Methodology, Resources, Software, Validation, Visualization, Writing – original draft. **Azad Abdullah Ameen:** Conceptualization, Methodology, Project administration, Resources, Supervision, Writing – review & editing.

Data availability: Data will be available upon reasonable request by the authors.

Conflicts of interest: The authors declare that they have no financial or non-financial conflicts of interest to disclose.

Funding: The authors did not receive support from any organization for the conducting of the study.

References

[1] T. Hamadneh, K. Kaabneh, O. Alssayed, K. Eguchi, Z. Monrazeri, and M. Dehghani, "Far and near optimization: a new simple and effective metaphor-less optimization algorithm for solving engineering applications," *CMES-Computer Modeling in Engineering & Sciences*, vol. 141, no. 2, pp. 1–10, 2024, doi: 10.32604/cmes.2024.053236.

[2] H. S. Abdulla, A. A. Ameen, S. I. Saeed, I. A. Mohammed, and T. A. Rashid, "MRSO: Balancing exploration and exploitation through modi-fied rat swarm optimization for global optimization," Algorithms. 2024; 17(9):423, doi: 10.3390/a17090423.

[3] H. M. Mohammed and T. A. Rashid, "FOX: A Fox-inspired optimization algorithm," *Applied Intelligence*, vol. 53, no. 1, pp. 1030–1050, 2023, doi: https://doi.org/10.21203/rs.3.rs-1939478/v1.

[4] D. O. Hasan and A. Aladdin, "Real-world applications of metaheuristic algorithms: a comprehensive review of the state-of-the-art," *Diyala Journal of Engineering Sciences*, pp. 1–27, Jun. 2025, doi: 10.24237/djes.2024.18201.

[5] D. Leiva, B. Ramos-Tapia, B. Crawford, R. Soto, and F. Cisternas-Caneo, "A novel approach to combinatorial problems: binary growth optimizer algorithm," *Biomimetics*, vol. 9, no. 5, May 2024, doi: 10.3390/biomimetics9050283.

[6] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, Aug. 2019, doi: 10.1016/j.future.2019.02.028.

[7] C. M. Rahman and T. A. Rashid, "A new evolutionary algorithm: learner performance based behavior algorithm," *Egyptian Informatics Journal*, vol. 22, no. 2, pp. 213–223, Jul. 2021, doi: 10.1016/j.eij.2020.08.003.

[8] E. Tuba, I. Strumberger, T. Bezdan, N. Bacanin, and M. Tuba, "Classification and feature selection method for medical datasets by brain storm optimization algorithm and support vector machine," in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 307–315. doi: 10.1016/j.procs.2019.11.289.

[9] A. S. Desuky, M. A. Cifci, S. Kausar, S. Hussain, and L. M. E. Bakrawy, "Mud ring algorithm: a new meta-heuristic optimization algorithm for solving mathematical and engineering challenges," *IEEE Access*, 2022, vol. 10, pp. 50448–50466, doi: 10.1109/ACCESS.2022.3173401.

[10] M. H. Zafar *et al.*, "A novel MPPT controller based on mud ring optimization algorithm for centralized thermoelectric generator under dynamic thermal gradients," *Applied Sciences (Switzerland)*, vol. 13, no. 7, Apr. 2023, doi: 10.3390/app13074213.

[11] G. Kulandaivelu, N. Verma, R. Arunachalam, and G. Muthusamy, "An innovative and adaptive deep Markov random field-based retailing prediction framework in big data analytics with improved mud ring optimizer," *Expert Systems with Applications*, vol. 270, 2025, p. 126489, doi: 10.1016/J.ESWA.2025.126489.

[12] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - *International Conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.

[13] R. R. Irshad *et al.*, "A multi-objective bee foraging learning-based particle swarm optimization algorithm for enhancing the security of healthcare data in cloud system," *IEEE Access*, vol. 11, pp. 113410–113421, 2023, doi: 10.1109/ACCESS.2023.3265954.

[14] Z. Shafiei Chafi and H. Afrakhte, "Short-term load forecasting using neural network and particle swarm optimization (pso) algorithm," *Mathematical Problems in Engineering*, vol. 2021, doi: 10.1155/2021/5598267.

[15] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[16] P. Hu, J. S. Pan, and S. C. Chu, "Improved binary grey wolf optimizer and its application for feature selection," *Knowledge-Based Systems*, vol. 195, May 2020, doi: 10.1016/j.knosys.2020.105746.

[17] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[18] R. Xu, C. Zhao, J. Li, J. Hu, and X. Hou, "A hybrid improved-whale-optimization–simulated-annealing algorithm for trajectory planning of quadruped robots," *Electronics (Switzerland)*, vol. 12, no. 7, Apr. 2023, doi: 10.3390/electronics12071564.

[19] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, Mar. 2016, doi: 10.1016/j.knosys.2015.12.022.

[20] T. ting Zhou and C. Shang, "Parameter identification of solar photovoltaic models by multi strategy sine–cosine algorithm," *Energy Science & Engineering*, vol. 12, no. 4, pp. 1422–1445, Apr. 2024, doi: 10.1002/ese3.1673.

[21] M. Ragab, A. A. Bahaddad, D. Hamed, A. Alkhayyat, D. Gupta, and R. F. Mansour, "Blockchain-driven privacy preserving electronic health records analysis using sine cosine algorithm with deep learning model," *Human-centric Computing and Information Sciences*, vol. 14, 2024, doi: 10.22967/HCIS.2024.14.009.

[22] 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," *Advances in Engineering Software*, vol. 114, pp. 163–191, Dec. 2017, doi: 10.1016/j.advengsoft.2017.07.002.

[23] Z. Yang, Y. Jiang, and W. C. Yeh, "Self-learning salp swarm algorithm for global optimization and its application in multi-layer perceptron model training," *Scientific Reports*, vol. 14, no. 1, p. 27401, Dec. 2024, doi: 10.1038/s41598-024-77440-4.

[24] M. S. Kiran, "TSA: Tree-seed algorithm for continuous optimization," *Expert Systems with Applications*, vol. 42, no. 19, pp. 6686–6698, May 2015, doi: 10.1016/j.eswa.2015.04.055.

[25] A. Beşkirli and İ. Dağ, "Parameter extraction for photovoltaic models with tree seed algorithm," *Energy Reports*, vol. 9, pp. 174–185, Mar. 2023, doi: 10.1016/j.egyr.2022.10.386.

[26] J. Liu, Y. Hou, Y. Li, and H. Zhou, "A multi-strategy improved tree–seed algorithm for numerical optimization and engineering optimization problems," *Scientific Reports*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-37958-5.

[27] A. S. Mashaleh, N. F. Binti Ibrahim, M. A. Al-Betar, H. M. J. Mustafa, and Q. M. Yaseen, "Detecting spam email with machine learning optimized with harris hawks optimizer (HHO) algorithm," in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 659–664. doi: 10.1016/j.procs.2022.03.087.

[28] D. T. Akl, M. M. Saafan, A. Y. Haikal, and E. M. El-Gendy, "IHHO: an improved harris hawks optimization algorithm for solving engineering problems," *Neural Computing and Applications*, vol. 36, no. 20, pp. 12185–12298, Jul. 2024, doi: 10.1007/s00521-024-09603-3.

[29] M. H. Qais, H. M. Hasanien, R. A. Turky, S. Alghuwainem, M. Tostado-Véliz, and F. Jurado, "Circle search algorithm: a geometry-based metaheuristic optimization algorithm," *Mathematics*, vol. 10, no. 10, May 2022, doi: 10.3390/math10101626.

[30] M. H. Qais, H. M. Hasanien, R. A. Turky, S. Alghuwainem, K. H. Loo, and M. Elgendy, "Optimal PEM fuel cell model using a novel circle search algorithm," *Electronics (Switzerland)*, vol. 11, no. 12, Jun. 2022, doi: 10.3390/electronics11121808.

[31] S. Talatahari and M. Azizi, "Chaos game optimization: a novel metaheuristic algorithm," *Artificial Intelligence Review*, vol. 54, no. 2, pp. 917–1004, Feb. 2021, doi: 10.1007/s10462-020-09867-w.

[32] A. Mabrouk, A. Dahou, M. A. Elaziz, R. P. Díaz Redondo, and M. Kayed, "medical image classification using transfer learning and chaos game optimization on the internet of medical things," *Computational Intelligence and Neuroscience*, vol. 2022, doi: 10.1155/2022/9112634.

[33] L. M. El Bakrawy, N. Bailek, L. Abualigah, S. Urooj, and A. S. Desuky, "Feature selection based on mud ring algorithm for improving survival prediction of children undergoing hematopoietic stem-cell transplantation," *Mathematics*, vol. 10, no. 22, Nov. 2022, doi: 10.3390/math10224197.

[34] S. Sivanesh, G. Mani, S. Venkatraman, and R. Nandhini, "Air quality prediction using ensemble voting based deep learning with mud ring algorithm for intelligent transportation systems," *Global Nest Journal*, vol. 25, no. 6, pp. 100–108, Aug. 2023, doi: 10.30955/gnj.004810.

[35] V. S. Kona, M. Subramoniam and V. A. Binson, "An efficient smart-IoT-aided waste management system with optimal shortest path routing using mud ring algorithm," *IEEE International Conference on Recent Advances in Systems, 2023*, doi: 10.1109/RASSE60029.2023.10363556.

[36] A. S. Alluhaidan, M. Maashi, M. A. Arasi, A. S. Salama, M. Assiri, and A. A. Alneil, "Mud ring optimization algorithm with deep learning model for disease diagnosis on ecg monitoring system," *Sensors*, vol. 23, no. 15, Aug. 2023, doi: 10.3390/s23156675.

[37] N. Maradona, and T. Jaya, "energy efficient trust routing based on mud 2 ring optimization in wireless sensor network 3." doi: https://dx.doi.org/10.2139/ssrn.4753854.

[38] M. SivaramKrishnan, R. Vidyalakshmi, R. Raguna, R. Shanmathi, R. Sharulatha, and K. Sujithra, "modulation of converter multiphase matrix for wind energy systems using a hybrid mud ring algorithm integrated with the elk herd

optimizer," *3rd International Conference on Automation, Computing and Renewable Systems, ICACRS 2024 - Proceedings*, pp. 277–283, 2024, doi: 10.1109/ICACRS62842.2024.10841773.

[39] K. J. Pradeep and P. K. Shukla, "Designing a novel network anomaly detection framework using multi-serial stacked network with optimal feature selection procedures over DDOS attacks," *International Journal of Intelligent Networks*, vol. 6, pp. 1–13, Jan. 2025, doi: 10.1016/j.ijin.2024.11.001.

[40] A. S. Desuky, S. Hussain, M. Akif Cifci, L. M. El Bakrawy, O. Mzoughi, and N. Kraiem, "parameter optimization based mud ring algorithm for improving the maternal health risk prediction," *IEEE Access*, vol. 12, pp. 167245–167261, 2024, doi: 10.1109/ACCESS.2024.3495518.

[41] A. Viktorin, R. Senkerik, M. Pluhacek, T. Kadavy and A. Zamuda, "Dish algorithm solving the CEC 2019 100-digit challenge," *2019 IEEE Congress on Evolutionary Computation (CEC),* Wellington, New Zealand, 2019, pp. 1-6, doi: 10.1109/CEC.2019.8789936.

[42] A. A. Ameen, T. A. Rashid, and S. Askar, "CDDO–HS: Child drawing development optimization–harmony search algorithm," *Applied Sciences*, vol. 13, no. 9, May 2023, doi: 10.3390/app13095795.

[43] A. A. Ameen, T. A. Rashid, and S. Askar, "MCDDO: overcoming challenges and enhancing performance in search optimization," Aug. 04, 2023. doi: 10.21203/rs.3.rs-3219594/v1.

[44] R. Eisinga, T. Heskes, B. Pelzer, and M. Te Grotenhuis, "Exact p-values for pairwise comparison of Friedman rank sums, with application to comparing classifiers," *BMC Bioinformatics*, vol. 18, no. 1, Jan. 2017, doi: 10.1186/s12859-017-1486-2.

[45] V. H. S. Pham, N. T. Nguyen Dang, and V. N. Nguyen, "Enhancing engineering optimization using hybrid sine cosine algorithm with Roulette wheel selection and opposition-based learning," *Scientific Reports*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-51343-w.

[46] G. Dhiman, "SSC: A hybrid nature-inspired meta-heuristic optimization algorithm for engineering applications," *Knowledge-Based Systems*, vol. 222, Jun. 2021, doi: 10.1016/j.knosys.2021.106926.

[47] E. S. M. El-Kenawy *et al.*, "Novel meta-heuristic algorithm for feature selection, unconstrained functions and engineering problems," *IEEE Access*, vol. 10, pp. 40536–40555, 2022, doi: 10.1109/ACCESS.2022.3166901.

[48] H. Bayzidi, S. Talatahari, M. Saraee, and C. P. Lamarche, "Social network search for solving engineering optimization problems," *Computational Intelligence and Neuroscience*, vol. 2021, 2021, doi: 10.1155/2021/8548639.

Appendix A

In this section, Tables A1 to A4 shows the mathematical expressions of the 23 widely recognized standard benchmark functions and CEC2019 set which employed and used to evaluate in this research. These tables offer an in-depth summary of the function equations, their dimensionality, specified upper and lower limit bounds, as well as their respective minimum target values. This detailed compilation forms the essential basis for assessing the effectiveness of the algorithms evaluated in the study [2, 41-43].

Table A1: Unimodal standard benchmark functions [2, 42].

Functions Formula	Dimension (n)	Range	f_{min}
$F_{1(x)} = \sum_{\{i=1\}}^{n} x_i^2$	30	[-100, 100]	0
$F_{2(x)} = \sum_{\{i=1\}}^{n} \lvert x_i \rvert + \prod_{\{i=1\}}^{n} x_i$	30	[-10, 10]	0
$F_{3(x)} = \sum_{\{i=1\}}^{n} \left(\sum_{\{j=1\}}^{n} x_j \right)^2$	30	[-100, 100]	0
$F_{4(x)} = max_{(\lvert x_i \rvert, 1 \le i \le n)}$	30	[-100, 100]	0
$F_{5(x)} = \sum_{i=1}^{n-1} \left(100(-x_i^2 + x_{i+1}) + (x_i - 1)\right)$	30	[-30, 30]	0
$F_{6(x)} = \sum_{\{i=1\}}^{n-1} ([x_i + 0.5])^2$	30	[-100, 100]	0
$F_{7(x)} = \sum_{\{i=1\}}^{n} i\, x_i^4 + rand[0,1]$	30	[-1.28, 1.28]	0

Table A2: Multimodal standard benchmark functions [2, 42]

Functions Formula	Dimension (n)	Range	f_{min}
$F_{8(x)} = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	[-500, 500]	-418.98
$F_{9(x)} = \sum_{i=1}^{n} [x_i^2 - 10\cos(\pi x_i) + 10]$	30	[-5.12, 5.12]	0
$F_{10(x)} = -20\, exp\left(^{-0.2}\sqrt{\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\,\pi x_i)\right) + 20 + e$	30	[-32, 32]	0
$F_{11(x)} = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600, 600]	0
$F_{12(x)} = \frac{\pi}{n}\Big\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin(\pi y_{i+1})^2] + (y_n - 1)^2\Big\} + \sum_{i=1}^{n} \mu(x_i, 10,100,4),\ y_i = 1 + \frac{x+1}{4}, \mu(x_i, a, k, m) = \begin{cases} k(x_i - a)^m x_i > a \\ 0 - a < x_i < a \\ k(-x_i - a)^m x_i < -a \end{cases}$	30	[-50, 50]	0

$F_{13(x)} = 0.1\left\{\sin(3\pi x_1)^2 + \sum_{i=1}^{n}(x_i-1)^2[1+\sin(3\pi x_i+1)^2] + (x_n-1)^2[\sin(2\pi x_n)^2]\right\} + \sum_{i=1}^{n}\mu(x_i,5,100,4)$	30	[-50, 50]	0

Table A3: Fixed-dimension Multimodal standard benchmark functions [2, 42]

Functions Formula	Dimension (n)	Range	f_{min}
$F_{14(x)} = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)$	2	[-65.536, 65.536]	1
$F_{15(x)} = \sum_{i=1}^{11}\left[a_i + \frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]$	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
$F_{17(x)} = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1-\frac{1}{8\pi}\right)\cos x_1 + 10$	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
$F_{19(x)} = -\sum_{i=1}^{4}c_i\,exp\left(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2\right)$	3	[0, 1]	-3.86
$F_{20(x)} = -\sum_{i=1}^{4}c_i\,exp\left(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2\right)$	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
$F_{23(x)} = -\sum_{i=1}^{10}0[(X-a_i)(X-a_i)^T+C_i]^{-1}$	4	[0, 10]	-10.5363

Table A4: CEC2019 benchmark functions [7, 41].

CEC	Functions	Dimension (n)	Range	f_{min}
01	Storn's Chebyshev polynomial fitting problem	9	[–8192, 8192]	1
02	Inverse Hilbert matrix problem	16	[–16384, 16384]	1
03	Lennard–Jones minimum energy cluster	18	[–4, 4]	1
04	Rastrigin's function	10	[–100, 100]	1
05	Griewangk's function	10	[–100, 100]	1
06	Weierstrass function	10	[–100, 100]	1
07	Modified Schwefel's function	10	[–100, 100]	1
08	Expanded Schaffer's F6 function	10	[–100, 100]	1
09	Happy Cat function	10	[–100, 100]	1
10	Ackley function	10	[–100, 100]	1