RESEARCH ARTICLE • OPEN ACCESS

A Hybrid Grey Wolf Optimizer–Zebra Optimization Algorithm for Solving Optimization Problems

Ayad Ali

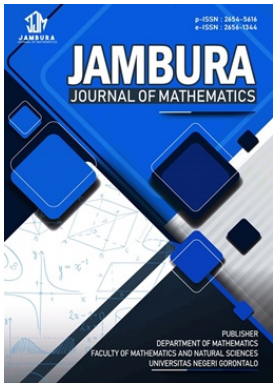
JAMBURA JOURNAL OF MATHEMATICS

Volume 8, Issue 1, Pages 7–25, February 2026

Received 19 September 2025, Revised 1 November 2025, Accepted 11 November 2025, Published 6 January 2026

To Cite this Article : A. Ali,"A Hybrid Grey Wolf Optimizer–Zebra Optimization Algorithm for Solving Optimization Problems", *Jambura J. Math*, vol. 8, no. 1, pp. 7–25, 2026, *https://doi.org/10.37905/jjom.v8i1.34499*

JOURNAL INFO • JAMBURA JOURNAL OF MATHEMATICS



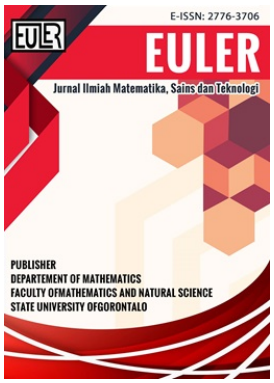
Homepage	:	http://ejurnal.ung.ac.id/index.php/jjom/index
Journal Abbreviation	:	Jambura J. Math.
Frequency	:	Biannual (February and August)
Publication Language	:	English (preferable), Indonesia
DOI	:	https://doi.org/10.37905/jjom
Online ISSN	:	2656-1344
Editor-in-Chief	:	Hasan S. Panigoro
Publisher	:	Department of Mathematics, Universitas Negeri Gorontalo
Country	:	Indonesia
OAI Address	:	http://ejurnal.ung.ac.id/index.php/jjom/oai
Google Scholar ID	:	iWLjgaUAAAAJ
Email	:	info.jjom@ung.ac.id

JAMBURA JOURNAL • FIND OUR OTHER JOURNALS

Jambura Journal of Biomathematics

Jambura Journal of Mathematics Education

Jambura Journal of Probability and Statistics



EULER : Jurnal Ilmiah Matematika, Sains, dan Teknologi

Jambura Journal of Mathematics, Volume 8, Issue 1, Pages 7–25, February 2026 https://doi.org/10.37905/jjom.v8i1.34499

Research Article

Check for updates

A Hybrid Grey Wolf Optimizer–Zebra Optimization Algorithm for Solving Optimization Problems

Ayad Ali[1,*]

[1]*Department of Mathematics, College of Science, University of Zakho, Zakho, Kurdistan Region, Iraq*

ARTICLE HISTORY
Received 19 September 2025
Revised 1 November 2025
Accepted 11 November 2025
Published 6 January 2026

KEYWORDS
Grey Wolf Optimizer
Zebra Optimization Algorithm
Hybrid Metaheuristic
Swarm Intelligence
Global Optimization

ABSTRACT. *Metaheuristic algorithms are widely applied to complex optimization problems, yet many suffer from premature convergence or slow search efficiency. To address these limitations, this paper proposes a new hybrid algorithm, Grey Wolf Optimizer–Zebra Optimization Algorithm (GWO–ZOA). The algorithm integrates the exploitation ability of the Grey Wolf Optimizer with the exploration capability of the Zebra Optimization Algorithm in a sequential framework, thereby enhancing both convergence accuracy and global search ability. The performance of GWO–ZOA is first evaluated on 23 standard benchmark functions, where it demonstrates competitive results in both unimodal and multimodal landscapes. Further validation is carried out on the CEC2017 and CEC2020 benchmark suites, confirming the hybrid's robustness across higher-dimensional and more challenging composite problems. In all three benchmark categories, the Friedman statistical test ranks GWO–ZOA first among the compared algorithms, highlighting its superior overall performance. Finally, the algorithm is applied to two real-world engineering design problems, where it consistently achieves high-quality feasible solutions and demonstrates practical effectiveness. These results confirm that the proposed GWO–ZOA algorithm is both robust and reliable for solving diverse and complex optimization tasks.*

1. Introduction

Optimization plays a vital role in science and engineering, as it involves searching for the most efficient or optimal solution among all possible alternatives. Many real-world problems in areas such as design, control, scheduling, and energy systems can be formulated as optimization tasks. Due to the increasing complexity and nonlinearity of modern problems, efficient optimization techniques have become essential tools for achieving accurate and reliable solutions.

Optimization algorithms can generally be categorized into deterministic and stochastic intelligent algorithms [1, 2]. Deterministic methods yield identical results under the same initial conditions, while stochastic algorithms rely on probabilistic mechanisms to explore the search space, often achieving better global performance on complex, multimodal problems. Among these, stochastic metaheuristic algorithms have gained significant attention due to their ability to balance exploration and exploitation when solving nonlinear and high-dimensional problems [3–5].

Over the past two decades, many nature-inspired metaheuristic algorithms have been introduced, modeled after biological and physical phenomena. Notable examples include the Particle Swarm Optimization (PSO) [6], Genetic Algorithm (GA) [7], Differential Evolution (DE) [8], Bat Algorithm (BA) [9], Firefly Algorithm (FA) [10, 11], Butterfly Optimization Algorithm (BOA) [12], Grey Wolf Optimizer (GWO) [13], Crow Search Algorithm (CSA) [14], and Three-group Exploration Strategy Algorithm (TGES) [15]. These algorithms have demonstrated competitive performance across various benchmark and engineering optimization tasks.

Among them, the Grey Wolf Optimizer (GWO) [13] is widely recognized for its simplicity, strong exploitation ability, and robust convergence behavior. It simulates the leadership hierarchy and cooperative hunting behavior of grey wolves. However, the standard GWO often exhibits limited exploration capability, which can lead to premature convergence, especially in complex or high-dimensional problems. Therefore, enhancing the exploration–exploitation balance of GWO remains an open research challenge.

The Zebra Optimization Algorithm (ZOA), proposed by Trojovská et al. [16], is another population-based metaheuristic inspired by the social behaviors of zebras, particularly their foraging and defensive mechanisms. ZOA exhibits a strong capacity for exploration through collective movement and maintains exploitation by modeling defensive strategies against predators. Although ZOA offers an effective balance between global and local search, there is still potential for improving its convergence rate and fine-tuning ability on complex optimization landscapes.

Numerous studies have focused on improving the original Grey Wolf Optimizer (GWO) and Zebra Optimization Algorithm (ZOA) by addressing their inherent limitations in exploration and convergence balance. For GWO, several enhanced versions have been proposed to mitigate premature convergence and improve population diversity. The Chaotic Grey Wolf Optimizer (CGWO) [17] employed chaotic maps to expand search coverage, while the Binary Grey Wolf Optimizer (BGWO) [18] adapted GWO for

*Corresponding Author.

Email : *ayad.ali@uoz.edu.krd* (A. Ali)
Homepage : http://ejurnal.ung.ac.id/index.php/jjom/index / E-ISSN : 2656-1344

discrete optimization tasks such as feature selection. Another variant, the Lévy flight–based GWO [20] incorporated Lévy flight dynamics to enhance both early exploration and late-slabele exploitation. Additionally, adaptive parameter-control approaches [21] and mutation-based variants [22] dynamically adjusted control parameters or introduced random perturbations to sustain exploration during slabelnation.

Similarly, the ZOA has also been refined to strengthen its global search efficiency. The Enhanced ZOA (EZOA) [23], which integrates Lévy flight and lens opposition-based learning, and the Chaotic ZOA (CZOA) [24], which employs chaotic maps, both improved exploration and population diversity. However, these improvements primarily emphasize exploration enhancement, often at the cost of slower convergence and reduced exploitation ability.

Beyond single-algorithm enhancements, several hybrid metaheuristics have been proposed to combine the strengths of multiple optimization strategies. Many of these methods merge GWO or ZOA with other established algorithms such as Particle Swarm Optimization (PSO) [25], Differential Evolution (DE) [19], or Whale Optimization Algorithm (WOA) [28] to improve adaptability. For instance, the Hybrid GWO–DE (HGWODE) [27] and GWO–WOA [28] algorithms employ symmetric or cooperative integration strategies, where parameters or position updates are shared between components. While such hybrids can enhance exploration, they frequently introduce additional computational complexity and lack a well-defined mechanism for maintaining an effective balance between exploration and exploitation.

In contrast, the unique novelty of the proposed GWO–ZOA hybrid lies in its strategic, asymmetrical integration: it specifically leverages the strong global exploitation of GWO to replace the initial ZOA foraging phase, and then selectively employs the defense-driven local search and occasional reset mechanism of ZOA. This deliberate, asymmetrical coupling directly addresses the complementary weaknesses of GWO's slabelnation and ZOA's over-exploration within a single, unified framework, offering a more effective and robust balance compared to typical, general-purpose hybridizations.

Given these observations, there exists a research gap in developing hybrid metaheuristic algorithms that can effectively combine the robust exploitation ability of GWO with the powerful exploration mechanisms of ZOA. Addressing this gap could lead to improved performance, higher accuracy, and faster convergence in solving diverse optimization problems. Therefore, the main objective of this study is to propose a hybrid Grey Wolf Optimizer–Zebra Optimization Algorithm (GWO–ZOA) that integrates the strengths of both algorithms to achieve a more balanced and efficient optimization performance. The proposed hybrid algorithm is evaluated on 23 classical benchmark functions, the CEC2017 and CEC2020 test suites, as well as several real-world engineering design problems to verify its effectiveness and practicality.

In summary, the structure of this paper is organized as follows. Section 2 presents the background of the Grey Wolf Optimizer (GWO) and Zebra Optimization Algorithm (ZOA), along with the detailed formulation of the proposed hybrid GWO–ZOA algorithm, including its motivation, operational mechanism, and overall framework. Section 3 provides a comprehensive evaluation of the proposed method using 23 classical benchmark functions, the IEEE CEC2017 and CEC2020 benchmark suites, and two real-world engineering design problems to verify its effectiveness and practical applicability. Finally, Section 4 summarizes the main findings and outlines potential directions for future research.

2. Methods

2.1. Problem definition and algorithmic background

In population-based metaheuristic optimization, a set of candidate solutions, referred to as agents, iteratively explores the search space to find the optimal solution for a given objective function. For a minimization problem, this can be formally defined as follows.

Find the vector $\mathbf{X} = \{x_1, x_2, \ldots, x_d\}$ such that

$$F(\mathbf{X}^*) = \min(F(\mathbf{X})), \quad \text{subject to } \mathbf{X} \in S,$$

where F is the objective function, d is the number of decision variables, $\mathbf{X}^*$ is the optimal solution, and S is the feasible region within the d-dimensional search space.

The population of N agents is represented by a matrix $\mathbf{X}$, where each row corresponds to an individual agent's position in the search space:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_N \end{bmatrix}_{N\times d} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,d} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,d} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,d} \end{bmatrix}_{N\times d}. \quad (1)$$

Here, $\mathbf{X}_i$ represents the i-th agent (e.g., a wolf or a zebra), and $x_{i,j}$ denotes the j-th dimension (decision variable) of its position.

The quality of each candidate solution is evaluated by the objective function. The corresponding fitness values for the entire population are stored in a vector $\mathbf{F}$:

$$\mathbf{F} = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(\mathbf{X}_1) \\ \vdots \\ F(\mathbf{X}_i) \\ \vdots \\ F(\mathbf{X}_N) \end{bmatrix}_{N\times 1}. \quad (2)$$

The goal of the optimization algorithms described in the following subsections is to evolve the population matrix $\mathbf{X}$ over successive iterations in order to minimize the values in the fitness vector $\mathbf{F}$, thereby converging toward the optimal solution $\mathbf{X}^*$.

2.2. Grey wolf optimization algorithm

The Grey Wolf Optimizer (GWO) [13] mimics the social hierarchy and cooperative hunting strategy of grey wolves. The population is divided into four ranks: alpha (α), beta (β), delta (δ), and omega (ω), as illustrated in Figure 1. The top three wolves (α, β, and δ) guide the search process, while the remaining wolves (ω) follow them, thereby maintaining population diversity.

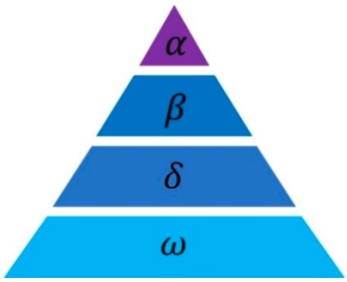


Figure 1. Leadership hierarchy in the Grey Wolf Optimizer (GWO)

The encircling behavior of wolves around prey is mathematically modeled as

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p^{\,t} - \vec{X}^{\,t} \right|, \tag{3}$$

$$\vec{X}^{\,t+1} = \vec{X}_p^{\,t} - \vec{A} \times \vec{D}, \tag{4}$$

where t represents the current number of iterations of the algorithm; $\vec{D}$ represents the distance between an individual grey wolf and its prey; $\vec{X}_p^{\,t}$ represents the position vector of the prey at time t; $\vec{X}^{\,t+1}$ represents the position vector of the grey wolf individual after updating its position; and $\vec{A}$ and $\vec{C}$ are positional parameter vectors. The position parameter vectors $\vec{A}$ and $\vec{C}$ are expressed as follows:

$$\vec{A} = 2a \cdot \vec{r}_1 - a, \tag{5}$$

$$\vec{C} = 2 \cdot \vec{r}_2, \tag{6}$$

$$a = 2\left(1 - \frac{t}{t_{\text{Max}}}\right), \tag{7}$$

where $\vec{A}$ is a parameter that decreases linearly with increasing iteration times; t_{Max} is the maximum number of iterations for the algorithm; $\vec{r}_1$ and $\vec{r}_2$ are random vectors distributed within the range $[0, 1]$; and a is used to balance the exploration and exploitation of the algorithm.

During hunting, the positions of α, β, and δ are used to guide other wolves:

$$\vec{D}_\alpha = \left| \vec{C}_1 \cdot \vec{X}_\alpha^{\,t} - \vec{X} \right|, \quad \vec{X}_1 = \vec{X}_\alpha^{\,t} - \vec{A}_1 \times \vec{D}_\alpha, \tag{8}$$

$$\vec{D}_\beta = \left| \vec{C}_2 \cdot \vec{X}_\beta^{\,t} - \vec{X} \right|, \quad \vec{X}_2 = \vec{X}_\beta^{\,t} - \vec{A}_2 \times \vec{D}_\beta, \tag{9}$$

$$\vec{D}_\delta = \left| \vec{C}_3 \cdot \vec{X}_\delta^{\,t} - \vec{X} \right|, \quad \vec{X}_3 = \vec{X}_\delta^{\,t} - \vec{A}_3 \times \vec{D}_\delta. \tag{10}$$

The final update rule is given by

$$\vec{X}^{\,t+1} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}, \tag{11}$$

where $\vec{D}_\alpha$, $\vec{D}_\beta$, and $\vec{D}_\delta$ represent the distances between the α, β, and δ wolves and their prey, respectively; $\vec{X}_\alpha^{\,t}$, $\vec{X}_\beta^{\,t}$, and $\vec{X}_\delta^{\,t}$ represent the positions of α, β, and δ at time t, respectively; $\vec{X}_1$, $\vec{X}_2$, and $\vec{X}_3$ represent the updated position vectors guided by α, β, and δ, respectively; and $\vec{A}_1$, $\vec{A}_2$, and $\vec{A}_3$ represent random vectors that affect the position update of α, β, and δ.

2.3. Zebra optimization algorithm

The Zebra Optimization Algorithm (ZOA) [16] is inspired by the social behaviors of zebras, particularly their foraging and defensive mechanisms. Each zebra represents a candidate solution whose position is updated based on the herd leader and its interaction with predators.

The following describes in detail the two natural behaviors of zebras considered in this algorithm:

- Phase 1: Foraging behavior
- Phase 2: Defense strategies against predators

2.3.1. Phase 1: Foraging behavior of the algorithm

One particular type of zebra is known as the plain zebra, which is considered a pioneer grazer. It feeds on higher, less nutritional grass canopy, creating an environment favorable for other species that require shorter grass. In the context of ZOA, the best individual within the population is called the *pioneer zebra*, which guides the rest of the population toward its location in the search space. Consequently, eq. (12)–(14) describe how zebras adjust their positions in the search space throughout the foraging phase:

$$x_{i,j}^{(t+1)} = x_{i,j}^t + r \cdot \left(PZ_j - I \cdot x_{i,j}^t\right), \tag{12}$$

$$I = \text{round}(1 + \text{rand}), \tag{13}$$

$$X_i^{(t+1)} = \begin{cases} X_i^{(t+1)}, & \text{if } F_i^{(t+1)} < F_i^t, \\ X_i^t, & \text{otherwise}, \end{cases} \tag{14}$$

where $X_i^{(t+1)}$ is the new state of the i-th zebra after the first phase, $x_{i,j}^{(t+1)}$ is the value of the j-th dimension for the i-th zebra, r is a random number in the interval $[0, 1]$, PZ_j is the j-th dimension of the pioneer zebra (best solution), rand is a random number in the interval $[0, 1]$, and $I \in \{1, 2\}$. If $I = 2$, this indicates larger changes in population movement.

2.3.2. Phase 2: Defending against predators

When threatened by predators, zebras must decide between two strategies: fleeing or defending. In the first strategy (S_1), the zebra selects an escape route in response to a lion's attack. In contrast, if a smaller predator (such as a dog) attacks, the zebra adopts an aggressive strategy (S_2) to scare it away. Accordingly, the movements of zebras in the search space during the defense phase can be described by eq. (15) and eq. (16):

$$x_{i,j}^{(t+1)} = \begin{cases} S_1: \; x_{i,j}^t + R \cdot (2r - 1)\left(1 - \dfrac{1}{t_{\text{max}}}\right) x_{i,j}^t, & \text{if } P_s \le 0.5, \\ S_2: \; x_{i,j}^t + r \cdot \left(AZ_j - I \cdot x_{i,j}^t\right), & \text{otherwise}, \end{cases} \tag{15}$$

$$X_i^{(t+1)} = \begin{cases} X_i^{(t+1)}, & \text{if } F_i^{(t+1)} < F_i^t, \\ X_i^t, & \text{otherwise}, \end{cases} \tag{16}$$

where $X_i^{(t+1)}$ is the new state of the i-th zebra after the second phase, R is a constant value equal to 0.01, P_s is the probability of selecting one of the two strategies and is randomly generated in the interval $[0, 1]$, and AZ_j is the j-th dimension of the attacked zebra.

2.4. Proposed hybrid GWO–ZOA algorithm

2.4.1. Motivation

Metaheuristic algorithms have been widely applied to solve complex and high-dimensional optimization problems due to their derivative-free nature, flexibility, and robustness across diverse problem domains [1]. However, a key challenge for most algorithms is achieving an effective balance between exploration and exploitation.

In contrast, the Grey Wolf Optimizer (GWO) [13] exhibits powerful exploitation ability through its leader-based hunting strategy, effectively refining solutions in promising regions of the search space. Nevertheless, GWO tends to suffer from a loss of diversity, which increases the risk of being trapped in local optima [22]. The Zebra Optimization Algorithm (ZOA) [16] demonstrates strong exploration capability through its biologically inspired foraging and defense mechanisms, which maintain diversity and reduce the risk of premature convergence in multimodal landscapes. Despite these advantages, ZOA's foraging step provides insufficient intensification, often resulting in slow convergence during later iterations.

To address these complementary weaknesses, this study proposes a hybrid GWO–ZOA algorithm in which the GWO hunting strategy is integrated to replace the exploratory ZOA foraging phase, thereby ensuring robust exploitation. Meanwhile, the ZOA defense phase is retained as a critical probabilistic mechanism to preserve population diversity.

In the original Zebra Optimization Algorithm (ZOA), the defense decision is governed by a fixed probability threshold of 0.5. This standard value creates an equiprobable choice between the exploration (flee) strategy and the exploitation (defend) strategy. However, since the hybrid GWO–ZOA already achieves strong exploitation through the GWO component, excessive exploration induced by ZOA's 50% flee probability can impede convergence. Therefore, the defense threshold is strategically reduced to 0.01. At each defense phase, a uniformly distributed random value $P_s \in [0, 1]$ is generated. The high-exploration escape strategy (S_1) is invoked only when $P_s \leq 0.01$; otherwise, the intensified local defense strategy (S_2) is applied.

This low-threshold configuration (0.01) was empirically determined to provide the most stable performance. Comparative analysis revealed that higher threshold values (e.g., 0.1–0.3), which are closer to the standard ZOA configuration, induced excessive defense-driven escapes, leading to over-exploration, slower convergence, and reduced solution accuracy. Conversely, employing a threshold of 0.01 effectively prioritizes the intensified local defense strategy (S_2), maintaining sufficient population diversity while leveraging the superior exploitation capability of GWO to achieve faster and more accurate convergence.

2.4.2. Work steps of the proposed hybrid algorithm

The development of the proposed hybrid algorithm followed a structured sequence of steps, which are summarized as follows.

Step 1: Initialization

A population of candidate solutions (wolves/zebras) is randomly generated within the defined search space. The problem dimension d, population size N, and maximum number of iterations $t_{\max}$ are set according to the benchmark functions or engineering problems under consideration. Each individual represents a potential solution to the optimization problem.

Step 2: Fitness Evaluation

The fitness values of all candidate solutions are evaluated, and the best solutions are identified. These fitness values are subsequently used to guide the update mechanisms in both the GWO and ZOA phases. At each iteration, the best solutions are retained to direct the search process in the subsequent iteration.

Step 3: Position Update by GWO-Based Exploitation (Replacing ZOA Foraging)

Instead of employing the original ZOA foraging mechanism, the population is updated using the GWO position update strategy. This step steers individuals toward promising regions of the search space based on the leadership hierarchy, thereby enhancing exploitation, accelerating convergence, and improving solution intensification.

Step 4: Position Update by ZOA Defense Phase (Exploration)

After the exploitation phase, the solutions undergo the ZOA defense mechanism to preserve population diversity. Unlike the original ZOA, which uses a fixed decision threshold of 0.5, the proposed hybrid algorithm applies a reduced probability threshold $P_s < 0.01$. With a very low probability, the escape strategy (S_1) is activated; otherwise, the intensified defensive strategy (S_2) is applied.

Step 5: Boundary Handling and Fitness Update

Any solution that exceeds the predefined search boundaries is corrected to remain within the allowable limits. The fitness values of the updated population are then recalculated, and the leading solutions are updated accordingly.

Step 6: Iteration and Stopping Condition

Steps 2–5 are repeated until the maximum number of iterations $t_{\max}$ is reached or a predefined convergence criterion is satisfied. The best solution obtained throughout the iterative process is finally reported as the global optimum.

2.4.3. Framework of GWO–ZOA algorithm

To clearly illustrate the operational mechanism of the proposed hybrid approach, this subsection presents the overall framework of the GWO–ZOA algorithm. The proposed method integrates the exploitation capability of the Grey Wolf Optimizer (GWO) with the exploration mechanism of the Zebra Optimization Algorithm (ZOA) in a structured and sequential manner. Specifically, the GWO hunting strategy is employed to guide the population toward promising regions of the search space, while the ZOA defense phase is retained to preserve population diversity and prevent premature convergence. The detailed procedural steps of the hybrid GWO–ZOA algorithm are summarized in Algorithm 1.

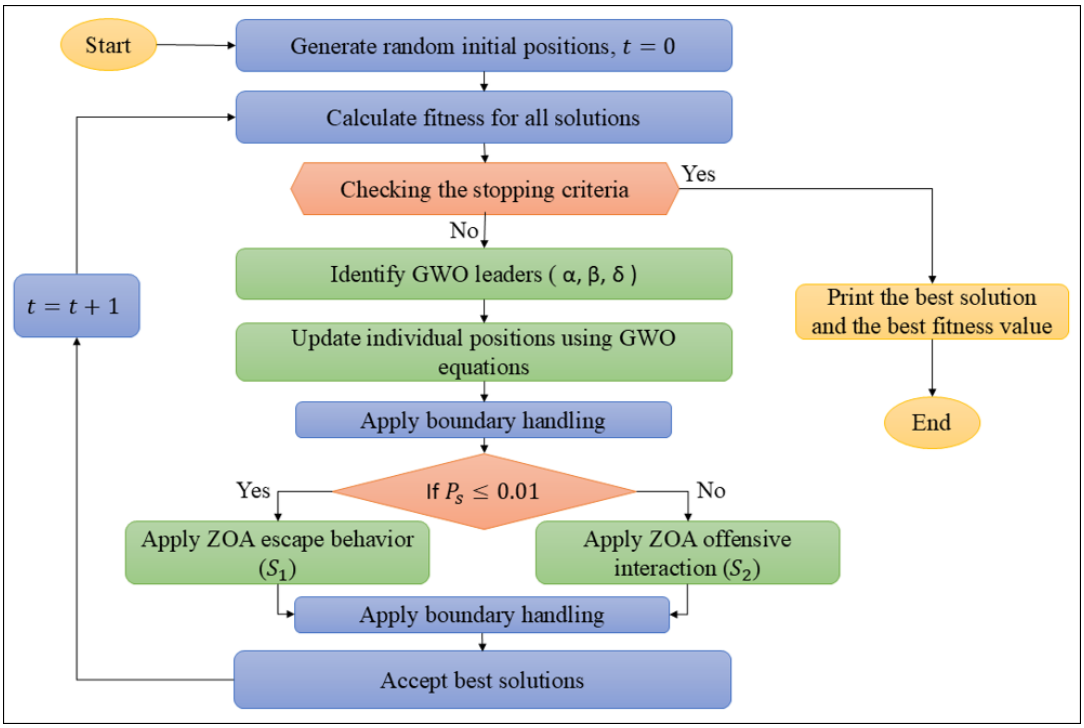


Figure 2. Flowchart of GWO-ZOA algorithm

Algorithm 1 Hybrid GWO–ZOA Algorithm

Input: Objective function $f(\mathbf{x})$, population size N, dimension d, lower bound **lb**, upper bound **ub**, maximum iterations $t_{\max}$
Output: Best solution $\mathbf{X}_{\text{best}}$ and best fitness f_{best}

1. Initialize population $\mathbf{X}_i$ for $i = 1, \ldots, N$ randomly within $[\mathbf{lb}, \mathbf{ub}]$
2. Set iteration counter $t = 0$
3. **Repeat while** $t < t_{\max}$:
   (a) Identify alpha (α), beta (β), and delta (δ) wolves as the three best solutions
   (b) Update all individuals using the GWO encircling and hunting strategy
   (c) Apply boundary control to ensure all solutions remain within $[\mathbf{lb}, \mathbf{ub}]$
   (d) For each individual, generate a random number $P_s \in (0, 1)$
   (e) If $P_s < 0.01$, perform ZOA escape behavior (Strategy S_1); otherwise, perform offensive or interaction behavior (Strategy S_2)
   (f) Apply boundary control again after the ZOA phase
   (g) Evaluate the fitness of all individuals using $f(\mathbf{x})$
   (h) Update $\mathbf{X}_{\text{best}}$ and f_{best} if a better solution is found
   (i) Increase iteration counter $t = t + 1$
4. **Return** $\mathbf{X}_{\text{best}}$ and f_{best}

For better visualization and understanding of the logical flow and interaction between the GWO exploitation phase and the ZOA defense phase, a flowchart representation of the proposed hybrid algorithm is provided. The flowchart illustrates the initialization process, iterative position updates, fitness evaluation, boundary handling, and termination conditions of the GWO–ZOA algorithm. The complete flowchart of the proposed method is shown in Figure 2.

2.4.4. Computational complexity analysis

The computational complexity of the proposed hybrid GWO–ZOA algorithm can be estimated based on the number of agents N, problem dimension d, and the maximum number of iterations $t_{\max}$. In the standard Grey Wolf Optimizer (GWO), the overall computational complexity per iteration is $\mathcal{O}(N \times d)$, which is mainly dominated by the position updating and fitness evaluation processes. Similarly, the Zebra Optimization Algorithm (ZOA) also exhibits a per-iteration complexity of $\mathcal{O}(N \times d)$.

In the proposed hybrid GWO–ZOA algorithm, the first phase of ZOA is replaced by the GWO position updating mechanism. Since both procedures involve updating N agents in a d-dimensional search space, the asymptotic computational complexity remains unchanged. Consequently, the total computational complexity of the hybrid algorithm is

$$\mathcal{O}(N \times d \times t_{\max}).$$

Although the hybridization introduces additional coefficient computations associated with the GWO mechanism, these operations consist of simple arithmetic calculations and do not significantly affect the overall computational cost. Therefore, the proposed hybrid GWO–ZOA algorithm preserves a computational complexity comparable to that of the original GWO and ZOA algorithms, while providing improved exploration–exploitation balance and enhanced solution quality.

2.4.5. Data analysis

The performance of the proposed hybrid algorithm is analyzed by evaluating its convergence speed, accuracy, and robustness in comparison with standard metaheuristic algorithms. The evaluation is conducted using classical benchmark functions, the CEC2017 and CEC2020 test suites, as well as several real-world engineering optimization problems.

Table 1. The parameters of the algorithms

Algorithm	Parameter	Value
PSO	inertia weight	Linearly decrease from 0.9 to 0.1
	c_1	2
	c_2	2
GWO	$\alpha_{\min}$ and $\alpha_{\max}$	0 and 2
SCA	a	Linearly decrease from 2 to 0
MFO	b	1
HHO	E_0	$[-1, 1]$
	E_1	Linearly decrease from 2 to 0
ZOA	p_s	Rand $\in [0, 1]$
	R	0.01

Table 2. Description of unimodal benchmark functions

Function	d	Range	$f_{\min}$
$F_1(x) = \sum_{i=1}^{d} x_i^2$	30	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^{d} \lvert x_i \rvert + \prod_{i=1}^{d} \lvert x_i \rvert$	30	$[-10, 10]$	0
$F_3(x) = \sum_{i=1}^{d} \left(\sum_{j=1}^{i} x_j \right)^2$	30	$[-100, 100]$	0
$F_4(x) = \max_i \{ \lvert x_i \rvert,\ 1 \le i \le n \}$	30	$[-100, 100]$	0
$F_5(x) = \sum_{i=1}^{d-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	30	$[-30, 30]$	0
$F_6(x) = \sum_{i=1}^{d} (\lvert x_i + 0.5 \rvert)^2$	30	$[-100, 100]$	0
$F_7(x) = \sum_{i=1}^{d} x_i^4 + \text{random}[0, 1)$	30	$[-1.28, 1.28]$	0

3. Results and Discussion

This section provides a detailed evaluation of the proposed hybrid algorithm. Its performance is comprehensively compared with six well-established metaheuristic algorithms, which are Particle Swarm Optimization (PSO) [6], Sine and Cosine Algorithm (SCA) [29], Moth-Flame Optimization (MFO) [30], Harris Hawk Optimization (HHO) [31], Grey Wolf Optimizer (GWO) [13], and Zebra Optimization Algorithm (ZOA) [16]. The experimental parameter values of all algorithms are shown in Table 1, which are selected from the original paper of each algorithm.

The evaluation is conducted over a diverse set of benchmark problems, including classical test functions, IEEE CEC benchmark suites, and constrained engineering design problems. The results are analyzed to highlight the optimization accuracy, convergence stability, and robustness of the proposed method in comparison with the reference algorithms.

3.1. Experimental setup

For all experiments, the population size was set to $N = 60$ and the maximum number of iterations was set to $t_{\max} = 700$, with the dimensionality of each benchmark function provided in the corresponding tables. Each function was independently executed for 30 runs, and the results were recorded using three performance metrics: the best fitness (BM), mean fitness (Mean), and standard deviation (STD). These metrics evaluate the optimization accuracy, average efficacy, and robustness of the algorithms.

All experiments were implemented using the MATLAB R2023a development environment. The computations were performed on a 64-bit ASUS laptop equipped with an Intel Core™ i5-4200H CPU @ 2.80 GHz processor, 12 GB of RAM, and the Windows 8.1 operating system.

3.2. Performance on 23 classical benchmark functions

In this section, 23 classical benchmark functions are selected for simulation experiments [32]. These functions are divided into unimodal functions and multimodal functions. Among them, F1–F7 are unimodal functions with only one global optimal solution, which are mainly used to test the optimization accuracy of the algorithm. The multimodal functions F8–F23 have multiple optima and are prone to falling into local optima. They are often used to test the exploration ability of the algorithm and its ability to avoid local optima. In addition, F14–F23 belong to fixed-dimensional multimodal functions, whose dimensions are lower and fixed; therefore, they have fewer local optimal solutions. The detailed mathematical formulations, dimensional settings, and search ranges of the unimodal and multimodal benchmark functions are summarized in Table 2–Table 4, while the comprehensive experimental results obtained by all compared algorithms on these 23 functions are reported in Table 5.

Table 3. Description of multimodal benchmark functions

Function	d	Range	$f_{\min}$
$F_8(x) = \sum_{i=1}^{d} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	$[-500, 500]$	$-418.9829\,d$
$F_9(x) = 10d + \sum_{i=1}^{d} \left[x_i^2 - 10\cos(2\pi x_i)\right]$	30	$[-5.12, 5.12]$	0
$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20 + \exp(1)$	30	$[-32, 32]$	0
$F_{11}(x) = \frac{1}{400}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0
$F_{12}(x) = \frac{\pi}{d}\left\{10\sin(\pi y_1) + \sum_{i=1}^{d}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_d - 1)^2\right\} + \sum_{i=1}^{d} u(x_i, 10, 100, 4),\ y_i = 1 + \frac{x_i + 1}{4},$ $u(x_i, a, i, n) = \begin{cases} k(x_i - a)^n, & x_i > a, \\ 0, & -a \le x_i \le a, \\ k(-x_i - a)^n, & x_i < -a \end{cases}$	30	$[-50, 50]$	0
$F_{13}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{d}(x_i - 1)^2\left[1 + \sin^2(3\pi x_i + 1)\right] + (x_d - 1)^2\left[1 + \sin^2(2\pi x_d)\right]\right\} + \sum_{i=1}^{d} u(x_i, 5, 100, 4)$ $u(x_i, a, i, n) = \begin{cases} k(x_i - a)^n, & x_i > a, \\ 0, & -a \le x_i \le a, \\ k(-x_i - a)^n, & x_i < -a \end{cases}$	30	$[-50, 50]$	0

Table 4. Description of fixed dimension multimodal benchmark functions

Function	d	Range	$f_{\min}$
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\left[\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right]\right)^{-1}$	2	$[-65.53, 65.53]$	0.998
$F_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	$[-5, 5]$	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1.0316
$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	2	$[-5, 10] \times [0, 15]$	0.398
$F_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right]$ $\left[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]$	2	$[-5, 5]$	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	$[0, 1]$	-3.22
$F_{21}(x) = -\sum_{i=1}^{5}\left[(X - a_i)(X - a_i)^T + 6c_i\right]^{-1}$	4	$[0, 10]$	-10.1532
$F_{22}(x) = -\sum_{i=1}^{7}\left[(X - a_i)(X - a_i)^T + 6c_i\right]^{-1}$	4	$[0, 10]$	-10.4029
$F_{23}(x) = -\sum_{i=1}^{10}\left[(X - a_i)(X - a_i)^T + 6c_i\right]^{-1}$	4	$[0, 10]$	-10.5364

As can be observed from Table 5, the proposed GWO–ZOA algorithm demonstrates superior overall performance compared with the other metaheuristics across the 23 benchmark functions. For the unimodal functions (F1–F7), GWO–ZOA consis-

Table 5. The comparison results of different algorithms on 23 benchmark functions

F	Stat	PSO	SCA	MFO	HHO	GWO	ZOA	GWO-ZOA
	BM	2.742E-11	2.574E-134	3.305E-03	2.761E-162	2.312E-57	0.000E+00	0.000E+00
F1	Mean	2.670E-03	1.524E-50	6.667E+02	8.191E-143	5.964E-55	0.000E+00	0.000E+00
	STD	8.224E-03	8.347E-50	2.537E+03	3.525E-142	1.282E-54	0.000E+00	0.000E+00
	BM	1.317E-03	7.424E-75	8.516E-03	1.246E-82	1.817E-33	2.477E-195	1.400E-234
F2	Mean	4.226E-02	6.551E-37	3.100E+01	4.632E-74	2.594E-32	2.035E-186	3.002E-230
	STD	4.525E-02	3.588E-36	1.807E+01	2.505E-73	2.800E-32	0.000E+00	0.000E+00
	BM	1.308E+02	7.466E+03	9.487E+02	1.313E-143	3.183E-13	5.930E-247	1.052E-289
F3	Mean	2.773E+02	4.784E+04	2.156E+04	3.864E-118	2.218E-08	9.111E-222	1.673E-265
	STD	1.136E+02	1.920E+04	1.443E+04	1.782E-117	6.891E-08	0.000E+00	0.000E+00
	BM	1.452E+00	6.015E+01	3.766E+01	1.842E-78	1.490E-12	4.996E-168	1.253E-203
F4	Mean	2.942E+00	7.572E+01	5.493E+01	4.093E-70	1.011E-10	1.875E-162	1.061E-198
	STD	7.273E-01	5.975E+00	1.079E+01	2.200E-69	1.384E-10	6.287E-162	0.000E+00
	BM	8.365E+00	2.870E+01	4.481E+01	1.982E-07	2.506E+01	2.854E+01	2.544E+01
F5	Mean	7.347E+01	1.039E+03	1.238E+04	1.763E-03	2.583E+01	2.878E+01	2.679E+01
	STD	5.417E+01	2.973E+03	3.099E+04	1.734E-03	5.842E-01	1.084E-01	9.155E-01
	BM	2.676E-12	9.512E-02	4.057E-03	1.730E-07	7.991E-06	2.743E+00	1.617E-05
F6	Mean	1.624E-03	6.493E-01	1.347E+03	1.193E-05	1.079E-01	3.730E+00	5.136E-01
	STD	4.679E-03	5.011E-01	3.492E+03	1.540E-05	1.369E-01	4.943E-01	2.747E-01
	BM	8.512E+00	8.336E+00	9.620E+00	7.420E+00	7.922E+00	7.698E+00	7.339E+00
F7	Mean	9.594E+00	9.695E+00	1.203E+01	8.633E+00	9.116E+00	8.439E+00	8.392E+00
	STD	5.334E-01	1.670E+00	3.413E+00	4.173E-01	4.690E-01	2.898E-01	4.054E-01
	BM	-8.166E+03	-1.257E+04	-1.014E+04	-1.257E+04	-8.533E+03	-7.409E+03	-7.807E+03
F8	Mean	-6.730E+03	-9.705E+03	-8.550E+03	-1.256E+04	-6.884E+03	-6.114E+03	-6.430E+03
	STD	7.254E+02	1.542E+03	7.439E+02	7.633E+01	1.240E+03	6.415E+02	6.347E+02
	BM	2.487E+01	0.000E+00	6.666E+01	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F9	Mean	4.998E+01	0.000E+00	1.499E+02	0.000E+00	9.884E+00	0.000E+00	0.000E+00
	STD	1.325E+01	0.000E+00	4.617E+01	0.000E+00	6.846E+00	0.000E+00	0.000E+00
	BM	1.282E-05	4.441E-16	2.094E-02	4.441E-16	4.441E-16	4.441E-16	4.441E-16
F10	Mean	1.122E-01	4.441E-16	1.021E+01	4.441E-16	3.405E-15	4.441E-16	4.441E-16
	STD	3.584E-01	0.000E+00	9.174E+00	0.000E+00	1.347E-15	0.000E+00	0.000E+00
	BM	5.950E-10	0.000E+00	7.686E-03	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F11	Mean	1.128E-02	0.000E+00	2.115E+01	0.000E+00	1.761E-03	0.000E+00	0.000E+00
	STD	1.656E-02	0.000E+00	4.557E+01	0.000E+00	4.114E-03	0.000E+00	0.000E+00
	BM	-1.022E+00	-9.086E-01	-1.021E+00	-1.020E+00	-1.022E+00	-9.328E-01	-1.012E+00
F12	Mean	-9.769E-01	1.130E+06	1.293E-01	-9.891E-01	-1.015E+00	-6.875E-01	-9.904E-01
	STD	1.472E-01	5.948E+06	1.355E+00	3.306E-02	5.972E-03	1.164E-01	1.328E-02
	BM	7.801E-09	1.552E-01	1.544E-02	1.746E-08	1.204E-05	1.245E+00	1.812E-01
F13	Mean	1.072E-02	1.331E+04	3.195E-01	1.087E-05	8.549E-02	1.975E+00	6.935E-01
	STD	1.576E-02	4.628E+04	5.854E-01	1.880E-05	9.625E-02	3.650E-01	2.265E-01
	BM	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01
F14	Mean	2.479E+00	1.163E+00	1.591E+00	1.097E+00	1.064E+00	2.445E+00	1.031E+00
	STD	2.134E+00	5.866E-01	1.385E+00	3.033E-01	3.622E-01	2.174E+00	1.815E-01
	BM	2.356E-04	4.275E-04	4.446E-04	4.487E-04	4.216E-04	2.374E-04	4.216E-04
F15	Mean	4.609E-04	6.448E-04	5.008E-04	5.625E-04	1.206E-03	1.323E-03	4.739E-04
	STD	6.154E-05	1.849E-04	1.174E-04	7.760E-05	3.998E-03	4.103E-03	1.458E-05
	BM	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00
F16	Mean	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00
	STD	6.775E-16	4.666E-05	6.775E-16	1.721E-11	4.531E-09	1.096E-11	1.889E-09
	BM	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01
F17	Mean	3.979E-01	3.983E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01
	STD	0.000E+00	1.032E-03	0.000E+00	1.369E-07	3.933E-05	9.999E-11	4.813E-08

tently achieves very high accuracy. In particular, the algorithm attains the best mean fitness values for F2, F3, F4, and F7, while tying with several methods on F1. This highlights the strong exploitation ability of the hybrid strategy, which enables it to refine solutions effectively and converge toward the global optimum with negligible variance. However, for F5 and F6, the GWO and HHO algorithms outperform GWO–ZOA. Nevertheless, the majority of unimodal results confirm that the hybridization significantly enhances convergence accuracy and stability.

For the multimodal functions (F8–F23), GWO–ZOA exhibits strong exploration capacity and robustness, although its performance varies across different landscapes. In F8, the hybrid is not competitive with HHO, SCA, MFO, GWO, and PSO, outperforming only ZOA. Similarly, in F12 and F13, GWO and PSO, respectively,

Table 5. The comparison results of different algorithms on 23 benchmark functions (Continued)

F	Stat	PSO	SCA	MFO	HHO	GWO	ZOA	GWO-ZOA
	BM	3.000E+00	3.000E+00	3.000E+00	3.000E+00	3.000E+00	3.000E+00	3.000E+00
F18	Mean	3.000E+00	3.738E+00	3.000E+00	3.000E+00	3.000E+00	3.000E+00	3.000E+00
	STD	1.125E-15	2.996E+00	1.428E-15	3.676E-09	1.345E-06	1.201E-06	6.272E-07
	BM	-3.863E+00	-3.863E+00	-3.863E+00	-3.863E+00	-3.863E+00	-3.863E+00	-3.863E+00
F19	Mean	-3.863E+00	-3.860E+00	-3.863E+00	-3.862E+00	-3.863E+00	-3.863E+00	-3.863E+00
	STD	2.710E-15	6.033E-03	2.710E-15	8.533E-04	1.178E-05	1.926E-04	2.468E-05
	BM	-3.322E+00	-3.321E+00	-3.322E+00	-3.299E+00	-3.322E+00	-3.322E+00	-3.322E+00
F20	Mean	-3.275E+00	-3.209E+00	-3.231E+00	-3.164E+00	-3.255E+00	-3.292E+00	-3.290E+00
	STD	5.940E-02	7.906E-02	5.147E-02	8.731E-02	7.143E-02	5.151E-02	5.403E-02
	BM	-1.015E+01	-1.010E+01	-1.015E+01	-1.012E+01	-1.015E+01	-1.015E+01	-1.015E+01
F21	Mean	-6.553E+00	-5.822E+00	-7.647E+00	-5.224E+00	-9.814E+00	-9.811E+00	-9.828E+00
	STD	3.525E+00	1.886E+00	3.415E+00	9.246E-01	1.288E+00	1.293E+00	1.238E+00
	BM	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01
F22	Mean	-8.365E+00	-6.704E+00	-9.129E+00	-5.611E+00	-1.023E+01	-9.694E+00	-1.040E+01
	STD	3.438E+00	2.483E+00	2.620E+00	1.599E+00	9.628E-01	1.838E+00	2.202E-04
	BM	-1.054E+01	-1.053E+01	-1.054E+01	-1.053E+01	-1.054E+01	-1.054E+01	-1.054E+01
F23	Mean	-7.330E+00	-6.114E+00	-8.221E+00	-5.484E+00	-9.995E+00	-1.036E+01	-1.054E+01
	STD	3.780E+00	2.123E+00	3.623E+00	1.353E+00	2.059E+00	9.873E-01	2.032E-04

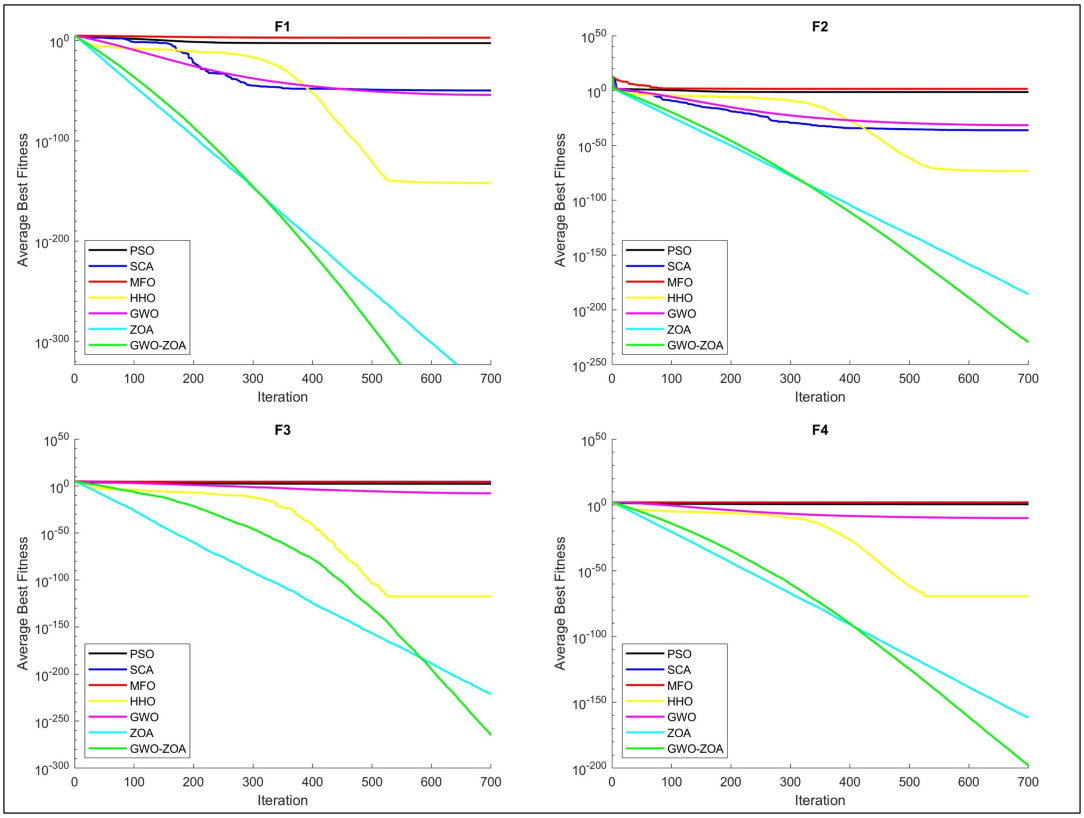


Figure 3. Convergence curves of the seven algorithms on unimodal benchmark functions (F1–F4)

obtain superior mean results, while in F15 the original GWO provides the best performance. In F20, ZOA slightly outperforms the hybrid. These cases suggest that, for certain rugged or deceptive landscapes, the mechanisms of the original algorithms can still provide an advantage over the hybridization. On the other hand, GWO–ZOA secures the best mean fitness values in F14 and in the complex multimodal functions F21–F23, where it significantly outperforms all competitors. Furthermore, in functions such as F9–F11 and F16–F19, all algorithms converge to the global optimum, with GWO–ZOA matching the best performance while maintaining extremely small standard deviations. These results confirm that the hybrid approach achieves a desirable balance between global exploration and local exploitation, ensuring both accuracy and stability across diverse problem landscapes.

Since the Friedman test was applied over 23 benchmark functions, the results reflect the overall ranking performance of each algorithm across all functions rather than on any single function. The significant p-value $(0.000168 < 0.05)$ confirms that at least one algorithm performed differently from the others in a statistically meaningful way. The average ranks across the 23

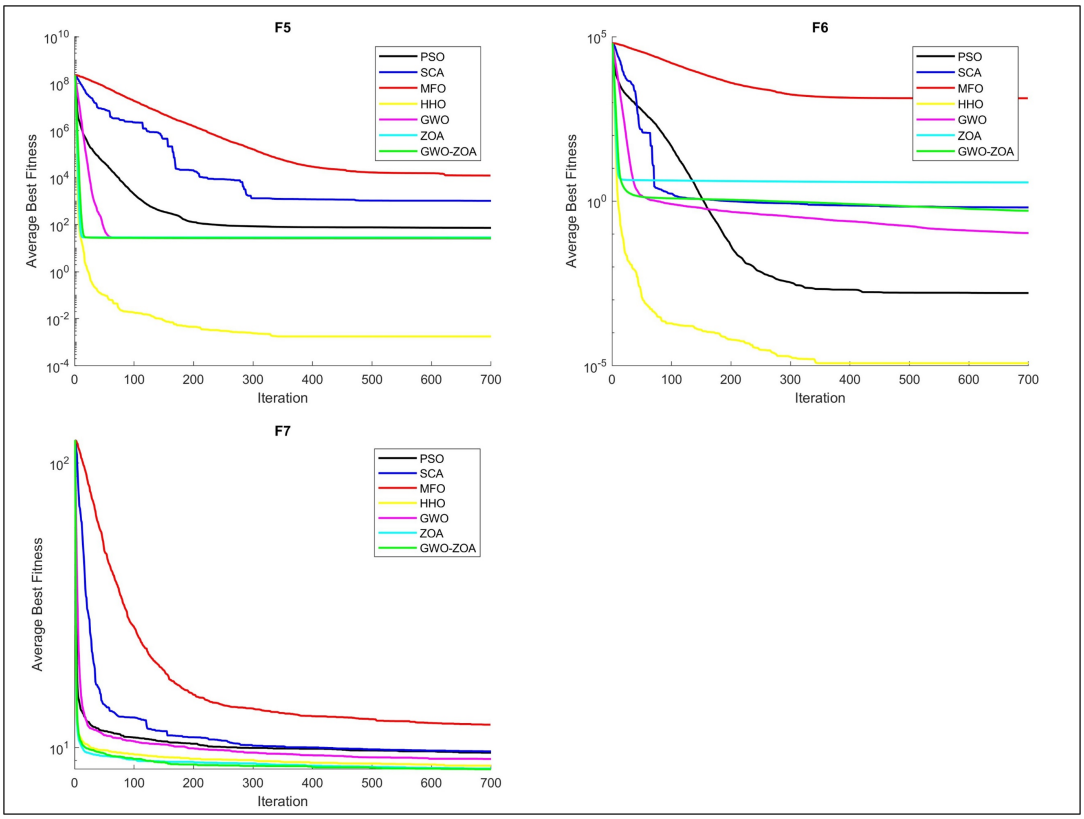


Figure 4. Convergence curves of the seven algorithms on unimodal benchmark functions (F5–F7)

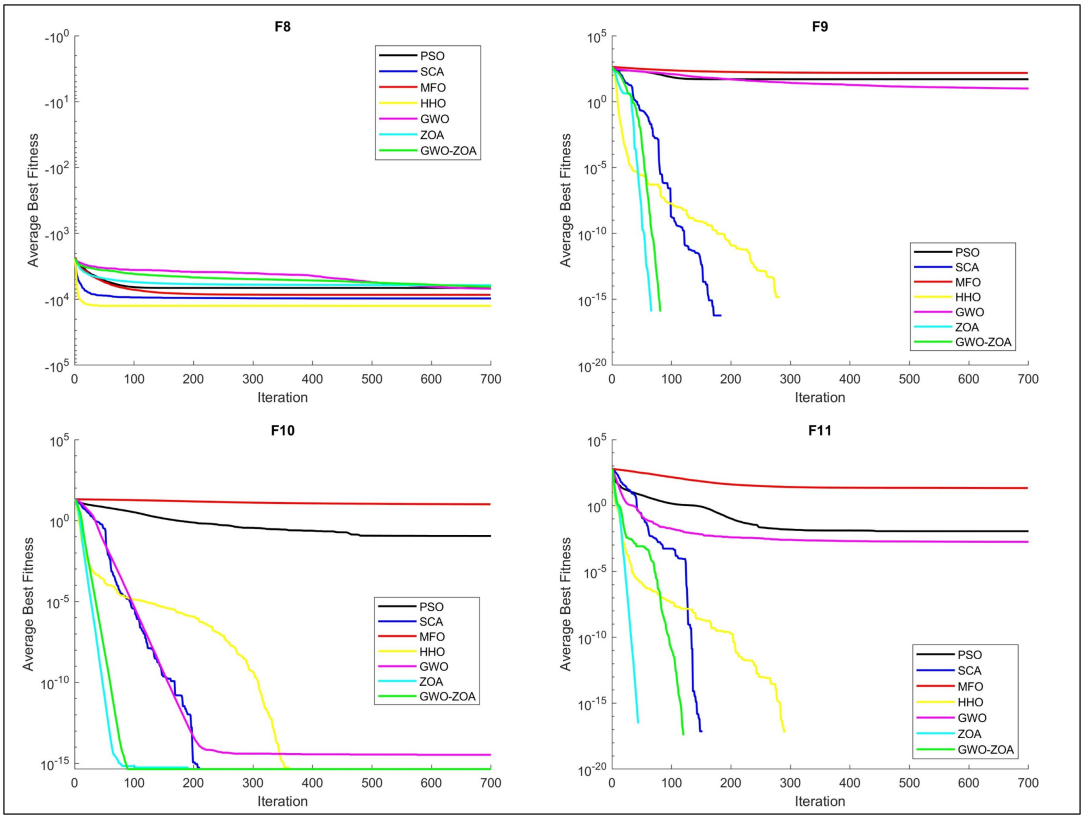


Figure 5. Convergence curves of the seven algorithms on multimodal benchmark functions (F8–F11)

functions are: PSO = 3.652, SCA = 5.043, MFO = 4.348, HHO = 3.217, GWO = 3.304, ZOA = 3.261, and GWO–ZOA = 2.174. Among these, the hybrid algorithm GWO–ZOA achieved the best performance with the lowest average rank, showing that it consistently outperformed the others across the majority of the 23 functions. In contrast, SCA obtained the highest rank (5.043),

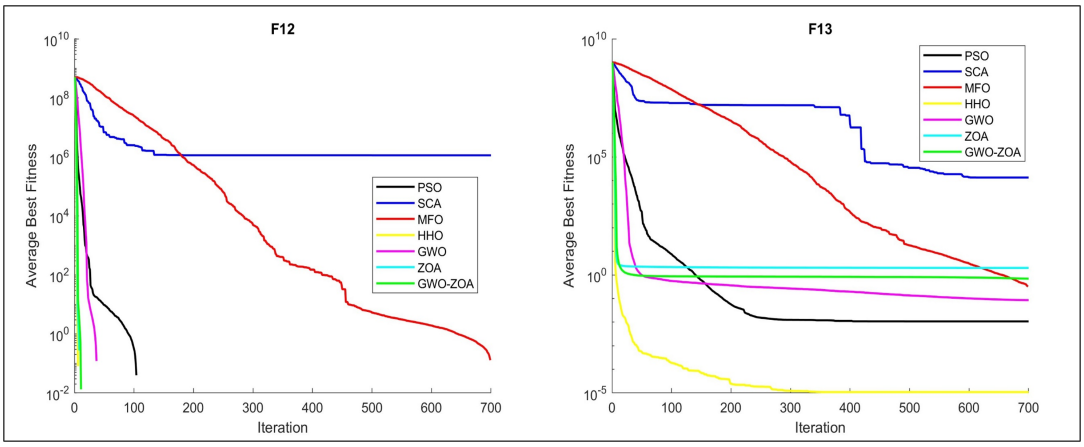


Figure 6. Convergence curves of the seven algorithms on multimodal benchmark functions (F12–F13)

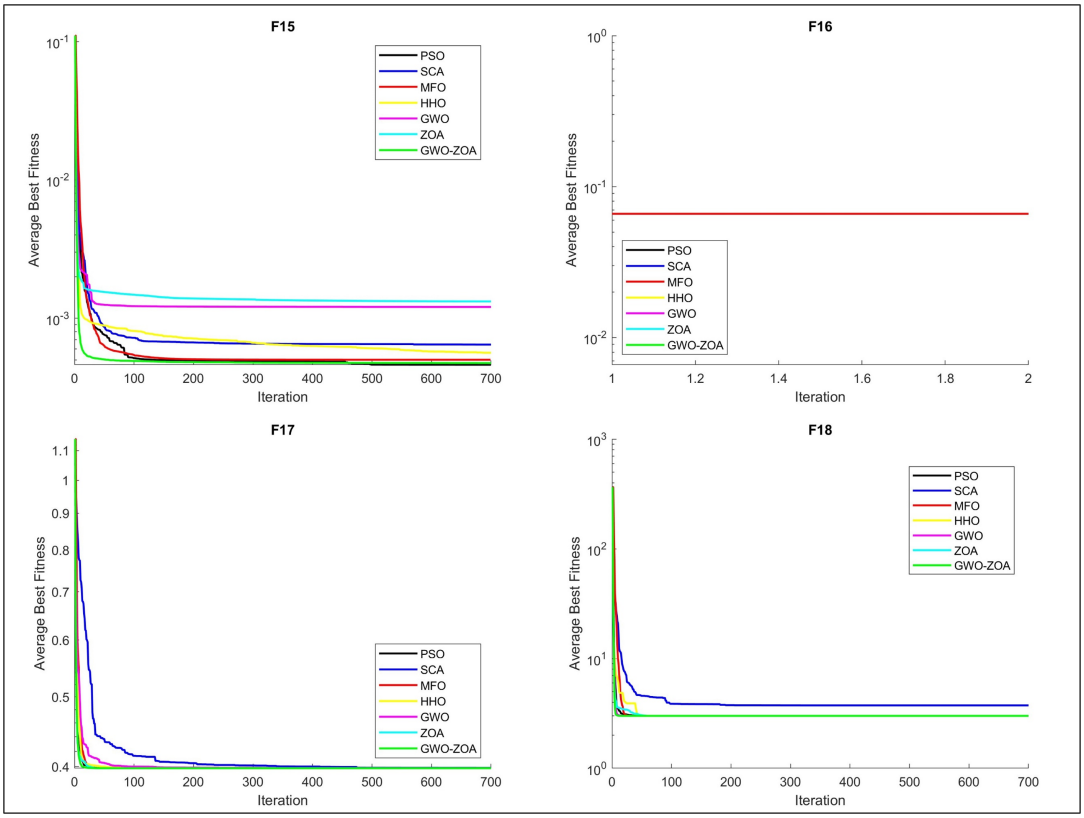


Figure 7. Convergence curves of the seven algorithms on fixed- dimension benchmark functions (F15–F18)

indicating the weakest overall performance. This demonstrates that the hybrid method is not only competitive but also statistically superior in solving a broad range of optimization problems.

To provide clearer insights into the optimization process, the convergence curves of all algorithms on the 23 standard benchmark functions are illustrated in Figure 3–Figure 8. In each plot, the vertical axis represents the average best fitness value obtained, while the horizontal axis denotes the number of iterations.

From Figure 3–Figure 8, the proposed GWO–ZOA demonstrates a faster convergence rate and more stable progression compared with most algorithms. On unimodal functions, it typically reaches near-optimal solutions more rapidly, highlighting its strong exploitation ability. On multimodal functions, the hybrid maintains steady convergence while reducing the risk of early stagnation, reflecting its capacity for controlled exploration. Although some algorithms achieve slightly better results on specific problems, the overall convergence behavior of GWO–ZOA indicates a well-balanced trade-off between speed and stability. These observations confirm its effectiveness as a competitive optimizer on the standard benchmark set.

3.3. *Performance on IEEE CEC2017 functions*

The circumstances of the CEC2017 test suite are more complex and demanding than those of unconstrained functions. Consequently, CEC2017 is chosen as the optimization problem for evaluating the GWO–ZOA algorithm, and the results are shown in Table 7. The CEC2017 test suite [33] comprises four categories of

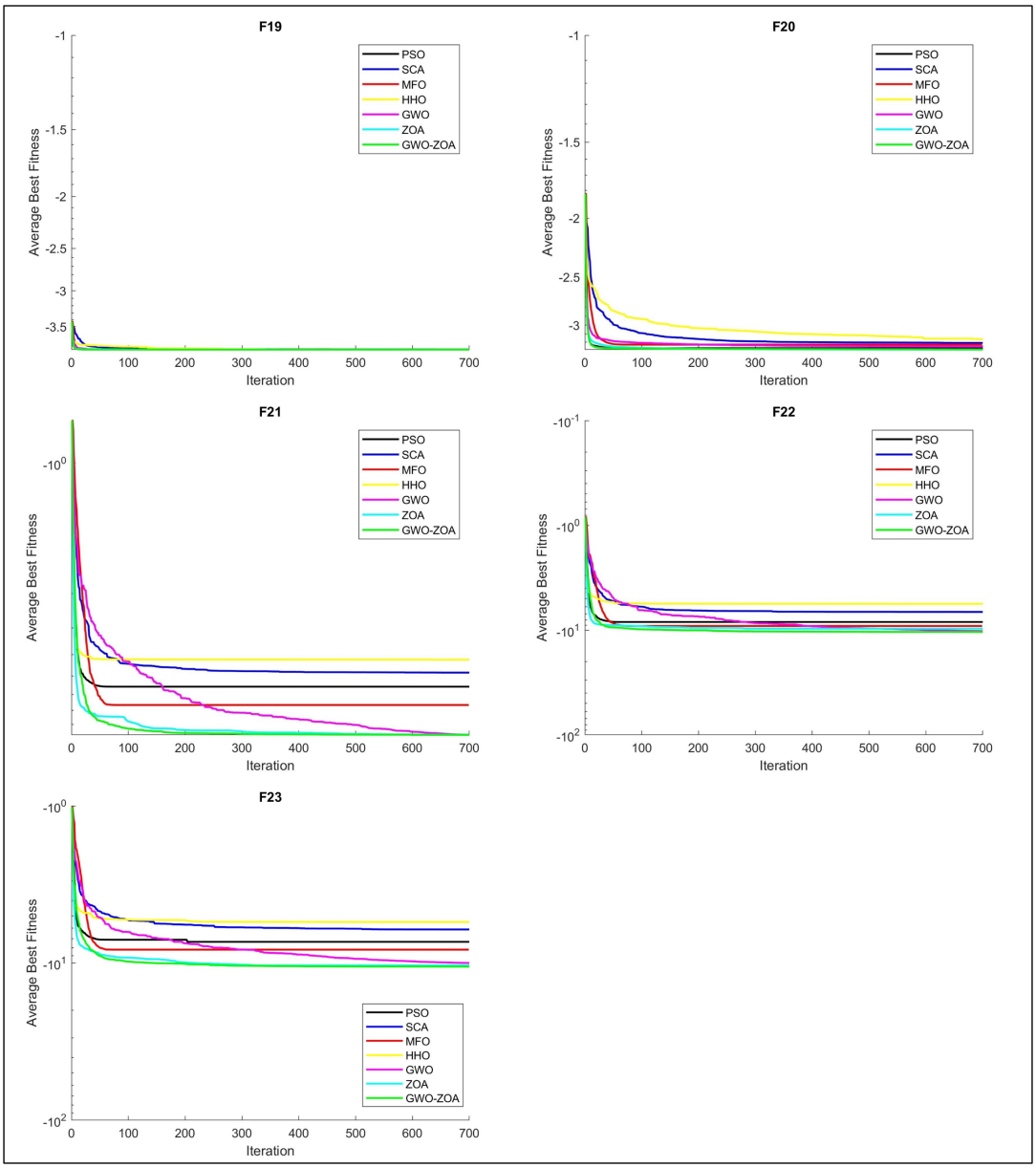


Figure 8. Convergence curves of the seven algorithms on fixed-dimension benchmark functions (F19–F23)

problems: F1–F3 are unimodal rotation displacement functions, typically employed to assess the convergence rate and optimization accuracy of algorithms; F4–F10 are multimodal rotation displacement functions, generally used to evaluate the algorithm's capacity to circumvent local optima. In addition, F11–F20 are hybrid functions, and F21–F30 are composition functions. It is difficult for most algorithms to reach the global optimal solution of hybrid and composition functions. Among them, F2 is deleted from the function list due to its instability and is not studied. Therefore, this section conducts simulation experiments based on 29 functions, and the specific information is shown in Table 6.

It is clear from Table 7 that the results on the CEC2017 benchmark set further confirm the superiority and robustness of the proposed GWO–ZOA algorithm. For several unimodal-like and composite functions such as C17-3, C17-5, C17-8, C17-10, C17-14, C17-16, C17-17, C17-18, and C17-20–C17-24, the hybrid achieves the best mean results among all competitors, clearly demonstrating its strong exploitation capability.

On the other hand, in some of the most difficult large-scale functions (such as C17-1, C17-4, C17-11, C17-12, C17-13, C17-15, C17-19, and C17-25–C17-30), the proposed algorithm does not achieve the best solutions. In these cases, algorithms such as PSO, HHO, or GWO outperform the hybrid.

Nevertheless, the Friedman test performed on the 29 CEC2017 benchmark functions shows a highly significant difference among the algorithms ($\chi^2 = 111.6946, p < 0.000001$). The average ranks of the algorithms are: PSO = 1.897, SCA = 6.621, MFO = 4.483, HHO = 4.586, GWO = 2.552, ZOA = 5.345, and GWO–ZOA = 2.517. Since lower ranks indicate better performance, PSO obtained the best overall performance, closely followed by the hybrid algorithm (GWO–ZOA = 2.517) and GWO (2.552). These three algorithms clearly form the top-performing group across the 29 benchmark functions. Meanwhile, SCA (6.621) recorded the weakest performance, while MFO (4.483),

Table 6. Description of CEC2017 benchmark functions

No.	Function	d	$f_{\min}$	No.	Function	d	$f_{\min}$
C17-1	Shifted and Rotated Bent Cigar Function	30	100	C17-16	Hybrid Function 6 ($N = 4$)	30	1600
C17-2	Shifted and Rotated Sum of Different Power Function	30	200	C17-17	Hybrid Function 6 ($N = 5$)	30	1700
C17-3	Shifted and Rotated Zakharov Function	30	300	C17-18	Hybrid Function 6 ($N = 5$)	30	1800
C17-4	Shifted and Rotated Rosenbrock's Function	30	400	C17-19	Hybrid Function 6 ($N = 5$)	30	1900
C17-5	Shifted and Rotated Rastrigin's Function	30	500	C17-20	Hybrid Function 6 ($N = 6$)	30	2000
C17-6	Shifted and Rotated Expanded Scaffer's Function	30	600	C17-21	Composition Function 1 ($N = 3$)	30	2100
C17-7	Shifted and Rotated Lunacek Bi-Rastrigin Function	30	700	C17-22	Composition Function 2 ($N = 3$)	30	2200
C17-8	Shifted and Rotated Non-Continuous Rastrigin's Function	30	800	C17-23	Composition Function 3 ($N = 4$)	30	2300
C17-9	Shifted and Rotated Levy Function	30	900	C17-24	Composition Function 4 ($N = 4$)	30	2400
C17-10	Shifted and Rotated Schwefel's Function	30	1000	C17-25	Composition Function 5 ($N = 5$)	30	2500
C17-11	Hybrid Function 1 ($N = 3$)	30	1100	C17-26	Composition Function 6 ($N = 5$)	30	2600
C17-12	Hybrid Function 2 ($N = 3$)	30	1200	C17-27	Composition Function 7 ($N = 6$)	30	2700
C17-13	Hybrid Function 3 ($N = 3$)	30	1300	C17-28	Composition Function 8 ($N = 6$)	30	2800
C17-14	Hybrid Function 4 ($N = 4$)	30	1400	C17-29	Composition Function 9 ($N = 3$)	30	2900
C17-15	Hybrid Function 5 ($N = 4$)	30	1500	C17-30	Composition Function 10 ($N = 3$)	30	3000

Table 7. The comparison results of different algorithms on CEC2017 functions with $D = 30$

F	Stat	PSO	SCA	MFO	HHO	GWO	ZOA	GWO-ZOA
C17-1	BM	1.191E+02	1.047E+10	1.184E+09	9.790E+06	1.023E+07	7.701E+09	1.826E+08
	Mean	4.406E+03	1.787E+10	7.508E+09	1.633E+07	2.970E+08	1.510E+10	2.018E+09
	STD	4.986E+03	4.526E+09	4.517E+09	3.253E+06	3.075E+08	4.598E+09	1.331E+09
C17-3	BM	2.720E+03	7.458E+04	3.032E+04	9.771E+03	2.427E+03	3.249E+04	9.966E+02
	Mean	8.331E+03	1.109E+05	1.047E+05	1.909E+04	7.668E+03	4.560E+04	5.355E+03
	STD	3.944E+03	2.052E+04	4.653E+04	5.115E+03	4.444E+03	7.232E+03	2.903E+03
C17-4	BM	4.004E+02	1.905E+03	4.938E+02	4.788E+02	4.854E+02	1.111E+03	5.029E+02
	Mean	4.915E+02	5.049E+03	7.792E+02	5.432E+02	5.527E+02	2.771E+03	5.905E+02
	STD	3.086E+01	2.328E+03	2.600E+02	4.570E+01	5.269E+01	9.170E+02	6.973E+01
C17-5	BM	5.488E+02	7.411E+02	5.982E+02	6.705E+02	5.401E+02	6.218E+02	5.395E+02
	Mean	6.043E+02	7.986E+02	6.796E+02	7.406E+02	6.176E+02	6.711E+02	5.763E+02
	STD	2.902E+01	3.955E+01	3.903E+01	3.493E+01	7.043E+01	3.399E+01	2.109E+01
C17-6	BM	6.004E+02	6.314E+02	6.087E+02	6.498E+02	6.004E+02	6.343E+02	6.105E+02
	Mean	6.081E+02	6.551E+02	6.247E+02	6.609E+02	6.031E+02	6.468E+02	6.183E+02
	STD	9.418E+00	1.200E+01	8.018E+00	5.792E+00	1.929E+00	6.249E+00	3.663E+00
C17-7	BM	7.664E+02	1.063E+03	8.306E+02	1.124E+03	7.807E+02	9.624E+02	7.880E+02
	Mean	8.000E+02	1.229E+03	9.882E+02	1.263E+03	8.724E+02	1.068E+03	8.454E+02
	STD	1.773E+01	9.817E+01	1.282E+02	7.537E+01	7.360E+01	7.677E+01	2.788E+01
C17-8	BM	8.478E+02	9.937E+02	9.294E+02	9.209E+02	8.394E+02	8.869E+02	8.376E+02
	Mean	8.896E+02	1.037E+03	9.926E+02	9.606E+02	8.830E+02	9.375E+02	8.624E+02
	STD	1.862E+01	2.506E+01	4.300E+01	2.293E+01	5.239E+01	2.836E+01	1.634E+01
C17-9	BM	9.099E+02	3.116E+03	2.119E+03	5.629E+03	9.544E+02	2.512E+03	1.142E+03
	Mean	2.068E+03	6.204E+03	6.558E+03	7.060E+03	1.340E+03	3.854E+03	1.548E+03
	STD	1.908E+03	1.780E+03	2.460E+03	9.021E+02	4.861E+02	6.283E+02	2.163E+02
C17-10	BM	2.908E+03	7.178E+03	3.954E+03	4.104E+03	2.577E+03	4.224E+03	2.694E+03
	Mean	4.390E+03	8.133E+03	5.516E+03	5.778E+03	5.854E+03	5.284E+03	3.701E+03
	STD	6.432E+02	4.101E+02	7.780E+02	6.750E+02	2.148E+03	6.476E+02	8.818E+02
C17-11	BM	1.174E+03	3.590E+03	1.359E+03	1.182E+03	1.194E+03	1.441E+03	1.208E+03
	Mean	1.230E+03	7.421E+03	3.269E+03	1.275E+03	1.272E+03	2.182E+03	1.350E+03
	STD	4.214E+01	2.211E+03	2.607E+03	4.984E+01	4.744E+01	4.783E+02	7.875E+01
C17-12	BM	9.846E+04	5.504E+08	3.105E+05	2.771E+06	1.159E+06	1.714E+07	4.793E+06
	Mean	6.306E+05	2.110E+09	1.122E+08	1.943E+07	2.511E+07	8.297E+08	4.395E+07
	STD	5.033E+05	1.154E+09	1.983E+08	1.684E+07	2.197E+07	7.086E+08	4.554E+07
C17-13	BM	2.134E+03	2.478E+06	1.320E+04	9.686E+04	3.271E+04	2.453E+04	2.127E+04
	Mean	1.467E+04	3.472E+08	2.904E+06	4.147E+05	8.638E+05	1.407E+08	2.240E+06
	STD	1.285E+04	6.007E+08	1.306E+07	2.275E+05	2.338E+06	2.996E+08	6.932E+06
C17-14	BM	2.209E+03	3.142E+04	4.986E+03	5.682E+03	3.540E+03	5.592E+03	1.682E+03
	Mean	2.788E+04	4.979E+05	1.748E+05	3.245E+05	6.325E+04	6.563E+05	2.679E+04
	STD	2.689E+04	5.376E+05	3.090E+05	3.517E+05	5.208E+04	5.234E+05	3.397E+04

Table 7. The comparison results of different algorithms on CEC2017 functions with $D = 30$ (Continued)

F	Stat	PSO	SCA	MFO	HHO	GWO	ZOA	GWO-ZOA
C17-15	BM	1.796E+03	1.678E+06	2.785E+03	1.829E+04	7.800E+03	9.601E+03	2.257E+04
	Mean	1.066E+04	5.847E+07	4.841E+04	6.149E+04	6.749E+04	3.381E+05	1.565E+05
	STD	9.928E+03	7.406E+07	4.585E+04	3.465E+04	5.331E+04	7.137E+05	3.201E+05
C17-16	BM	1.911E+03	2.848E+03	2.413E+03	2.505E+03	1.829E+03	2.443E+03	1.845E+03
	Mean	2.416E+03	3.687E+03	3.056E+03	3.214E+03	2.374E+03	3.067E+03	2.284E+03
	STD	2.753E+02	4.740E+02	2.981E+02	4.304E+02	3.434E+02	3.278E+02	2.812E+02
C17-17	BM	1.776E+03	2.160E+03	2.126E+03	1.825E+03	1.790E+03	1.865E+03	1.794E+03
	Mean	1.998E+03	2.450E+03	2.446E+03	2.569E+03	1.956E+03	2.294E+03	1.930E+03
	STD	1.355E+02	2.167E+02	2.028E+02	2.824E+02	1.526E+02	2.454E+02	1.112E+02
C17-18	BM	3.958E+04	1.592E+05	8.739E+04	7.210E+04	7.606E+04	7.326E+04	5.286E+04
	Mean	4.833E+05	3.018E+06	2.539E+06	1.480E+06	8.847E+05	2.565E+06	3.623E+05
	STD	4.834E+05	3.721E+06	5.641E+06	1.410E+06	7.868E+05	3.591E+06	3.659E+05
C17-19	BM	1.957E+03	1.526E+06	2.201E+03	6.407E+04	1.580E+04	1.017E+05	2.814E+04
	Mean	1.042E+04	3.948E+07	7.575E+05	4.955E+05	2.906E+05	1.715E+06	7.904E+05
	STD	1.292E+04	4.621E+07	2.850E+06	4.801E+05	6.823E+05	1.624E+06	8.334E+05
C17-20	BM	2.180E+03	2.490E+03	2.097E+03	2.338E+03	2.105E+03	2.267E+03	2.109E+03
	Mean	2.409E+03	2.700E+03	2.533E+03	2.714E+03	2.323E+03	2.443E+03	2.275E+03
	STD	1.833E+02	1.378E+02	2.421E+02	2.412E+02	1.501E+02	1.145E+02	6.099E+01
C17-21	BM	2.351E+03	2.506E+03	2.414E+03	2.430E+03	2.342E+03	2.412E+03	2.331E+03
	Mean	2.392E+03	2.569E+03	2.482E+03	2.551E+03	2.392E+03	2.474E+03	2.362E+03
	STD	2.756E+01	3.510E+01	3.575E+01	4.892E+01	4.672E+01	3.424E+01	1.489E+01
C17-22	BM	2.300E+03	3.607E+03	2.417E+03	2.327E+03	2.320E+03	3.496E+03	2.403E+03
	Mean	3.499E+03	7.195E+03	5.598E+03	6.265E+03	4.923E+03	5.230E+03	2.669E+03
	STD	1.886E+03	2.008E+03	1.866E+03	1.871E+03	2.748E+03	1.014E+03	2.189E+02
C17-23	BM	2.712E+03	2.940E+03	2.762E+03	2.941E+03	2.690E+03	2.934E+03	2.686E+03
	Mean	2.782E+03	3.132E+03	2.814E+03	3.126E+03	2.755E+03	3.124E+03	2.726E+03
	STD	4.804E+01	9.190E+01	2.930E+01	9.449E+01	6.367E+01	9.263E+01	2.546E+01
C17-24	BM	2.885E+03	3.133E+03	2.909E+03	3.223E+03	2.850E+03	3.180E+03	2.845E+03
	Mean	2.933E+03	3.266E+03	2.968E+03	3.375E+03	2.948E+03	3.389E+03	2.873E+03
	STD	2.964E+01	1.168E+02	2.673E+01	1.240E+02	7.933E+01	1.025E+02	1.454E+01
C17-25	BM	2.884E+03	3.284E+03	2.889E+03	2.891E+03	2.892E+03	3.091E+03	2.921E+03
	Mean	2.897E+03	3.594E+03	3.103E+03	2.929E+03	2.933E+03	3.365E+03	2.985E+03
	STD	1.418E+01	2.092E+02	2.397E+02	2.115E+01	1.993E+01	1.927E+02	4.095E+01
C17-23	BM	2.712E+03	2.940E+03	2.762E+03	2.941E+03	2.690E+03	2.934E+03	2.686E+03
	Mean	2.782E+03	3.132E+03	2.814E+03	3.126E+03	2.755E+03	3.124E+03	2.726E+03
	STD	4.804E+01	9.190E+01	2.930E+01	9.449E+01	6.367E+01	9.263E+01	2.546E+01
C17-24	BM	2.885E+03	3.133E+03	2.909E+03	3.223E+03	2.850E+03	3.180E+03	2.845E+03
	Mean	2.933E+03	3.266E+03	2.968E+03	3.375E+03	2.948E+03	3.389E+03	2.873E+03
	STD	2.964E+01	1.168E+02	2.673E+01	1.240E+02	7.933E+01	1.025E+02	1.454E+01
C17-25	BM	2.884E+03	3.284E+03	2.889E+03	2.891E+03	2.892E+03	3.091E+03	2.921E+03
	Mean	2.897E+03	3.594E+03	3.103E+03	2.929E+03	2.933E+03	3.365E+03	2.985E+03
	STD	1.418E+01	2.092E+02	2.397E+02	2.115E+01	1.993E+01	1.927E+02	4.095E+01
C17-26	BM	2.800E+03	6.532E+03	4.841E+03	2.953E+03	3.657E+03	5.269E+03	3.700E+03
	Mean	3.896E+03	7.891E+03	5.513E+03	7.082E+03	4.313E+03	7.436E+03	4.768E+03
	STD	1.145E+03	6.424E+02	4.132E+02	1.540E+03	2.089E+02	6.549E+02	5.023E+02
C17-27	BM	3.205E+03	3.256E+03	3.216E+03	3.281E+03	3.203E+03	3.544E+03	3.222E+03
	Mean	3.245E+03	3.554E+03	3.237E+03	3.429E+03	3.224E+03	3.826E+03	3.256E+03
	STD	3.081E+01	1.476E+02	1.579E+01	1.527E+02	1.080E+01	1.438E+02	1.797E+01
C17-28	BM	3.189E+03	3.784E+03	3.314E+03	3.254E+03	3.240E+03	3.887E+03	3.312E+03
	Mean	3.218E+03	4.951E+03	3.798E+03	3.305E+03	3.314E+03	4.371E+03	3.413E+03
	STD	1.732E+01	4.405E+02	5.911E+02	3.145E+01	4.460E+01	3.439E+02	6.186E+01
C17-29	BM	3.428E+03	4.194E+03	3.611E+03	4.123E+03	3.401E+03	3.988E+03	3.522E+03
	Mean	3.674E+03	4.887E+03	4.012E+03	4.513E+03	3.667E+03	4.997E+03	3.749E+03
	STD	1.731E+02	4.096E+02	2.507E+02	2.788E+02	2.000E+02	5.320E+02	1.615E+02
C17-30	BM	5.855E+03	1.103E+07	9.046E+03	1.252E+05	8.692E+05	5.744E+05	1.416E+06
	Mean	9.475E+03	6.130E+07	4.413E+05	2.434E+06	4.961E+06	1.414E+07	7.365E+06
	STD	3.321E+03	5.888E+07	1.091E+06	1.636E+06	2.687E+06	1.222E+07	4.288E+06

Table 8. Description of CEC2020 benchmark functions

No.	Function	d	$f_{\min}$
C20-1	Shifted and Rotated Bent Cigar Function	10	100
C20-2	Shifted and Rotated Schwefel's Function	10	1100
C20-3	Shifted and Rotated Lunacek Bi-Rastrigin Function	10	700
C20-4	Expanded Rosenbrock's plus Griewangk's Function	10	1900
C20-5	Hybrid Function ($\mu = 3$)	10	1700
C20-6	Hybrid Function ($\mu = 4$)	10	1600
C20-7	Hybrid Function ($\mu = 5$)	10	2100
C20-8	Composition Function ($\mu = 3$)	10	2200
C20-9	Composition Function ($\mu = 4$)	10	2400
C20-10	Composition Function ($\mu = 5$)	10	2500

Table 9. The comparison results of different algorithms on CEC2020 functions

F	Stat	PSO	SCA	MFO	HHO	GWO	ZOA	GWO-ZOA
C20-1	BM	1.146E+03	2.313E+09	1.748E+03	3.484E+05	1.712E+04	3.861E+08	5.082E+04
	Mean	1.146E+03	2.313E+09	1.748E+03	3.484E+05	1.712E+04	3.861E+08	5.082E+04
	STD	2.313E-13	1.455E-06	6.938E-13	5.920E-11	7.400E-12	6.062E-08	3.700E-11
C20-2	BM	1.689E+03	2.033E+03	1.909E+03	1.914E+03	1.510E+03	1.759E+03	1.120E+03
	Mean	1.689E+03	2.033E+03	1.909E+03	1.914E+03	1.510E+03	1.759E+03	1.120E+03
	STD	2.313E-13	6.938E-13	6.938E-13	6.938E-13	6.938E-13	9.250E-13	6.938E-13
C20-3	BM	7.258E+02	7.545E+02	7.346E+02	7.755E+02	7.272E+02	7.381E+02	7.164E+02
	Mean	7.258E+02	7.545E+02	7.346E+02	7.755E+02	7.272E+02	7.381E+02	7.164E+02
	STD	0.000E+00	1.156E-13	5.782E-13	3.469E-13	3.469E-13	1.156E-13	2.313E-13
C20-4	BM	1.901E+03	1.900E+03	1.901E+03	1.900E+03	1.901E+03	1.900E+03	1.900E+03
	Mean	1.901E+03	1.900E+03	1.901E+03	1.900E+03	1.901E+03	1.900E+03	1.900E+03
	STD	1.156E-12	0.000E+00	0.000E+00	0.000E+00	2.313E-13	0.000E+00	0.000E+00
C20-5	BM	3.462E+03	1.479E+04	6.566E+03	3.721E+03	4.878E+03	4.827E+04	9.776E+03
	Mean	3.462E+03	1.479E+04	6.566E+03	3.721E+03	4.878E+03	4.827E+04	9.776E+03
	STD	2.313E-12	5.550E-12	1.850E-12	2.313E-12	9.250E-13	1.480E-11	0.000E+00
C20-6	BM	1.851E+03	1.610E+03	1.785E+03	1.733E+03	1.769E+03	1.858E+03	1.729E+03
	Mean	1.851E+03	1.610E+03	1.785E+03	1.733E+03	1.769E+03	1.858E+03	1.729E+03
	STD	6.938E-13	2.313E-13	1.156E-12	9.250E-13	6.938E-13	4.625E-13	6.938E-13
C20-7	BM	2.409E+03	5.614E+04	6.543E+03	5.994E+03	2.705E+03	8.950E+03	2.699E+03
	Mean	2.409E+03	5.614E+04	6.543E+03	5.994E+03	2.705E+03	8.950E+03	2.699E+03
	STD	9.250E-13	2.960E-11	0.000E+00	3.700E-12	1.850E-12	7.400E-12	0.000E+00
C20-8	BM	2.301E+03	2.328E+03	2.301E+03	2.311E+03	2.308E+03	2.381E+03	2.309E+03
	Mean	2.301E+03	2.328E+03	2.301E+03	2.311E+03	2.308E+03	2.381E+03	2.309E+03
	STD	0.000E+00	0.000E+00	1.850E-12	9.250E-13	4.625E-13	1.850E-12	1.850E-12
C20-9	BM	2.500E+03	2.580E+03	2.749E+03	2.501E+03	2.736E+03	2.506E+03	2.735E+03
	Mean	2.500E+03	2.580E+03	2.749E+03	2.501E+03	2.736E+03	2.506E+03	2.735E+03
	STD	0.000E+00	0.000E+00	1.388E-12	1.850E-12	9.250E-13	1.850E-12	1.388E-12
C20-10	BM	2.899E+03	2.965E+03	2.948E+03	2.946E+03	2.948E+03	2.963E+03	2.921E+03
	Mean	2.899E+03	2.965E+03	2.948E+03	2.946E+03	2.948E+03	2.963E+03	2.921E+03
	STD	9.250E-13	4.625E-13	0.000E+00	1.388E-12	4.625E-13	1.388E-12	1.850E-12

HHO (4.586), and ZOA (5.345) showed moderate results. Overall, the findings confirm that the hybrid algorithm performs at a highly competitive level, ranking second overall and demonstrating strong robustness across the CEC2017 test suite.

3.4. Performance on IEEE CEC2020 functions

The CEC2020 [34] competition on bound-constrained numerical optimization problems is used to test the performance of the GWO–ZOA algorithm, and the results are shown in Table 9. This benchmark set consists of 10 optimization problems referred to as C20-1 to C20-10. In summary, Function 1 is unimodal, Functions 2–4 are basic functions, Functions 5–7 are hybrid functions, and Functions 8–10 are composition functions, as shown in Table 8. Thus, in these problems, finding the best solution is a challenging task due to multi-modality, hybrid, and composition features. More details can be found in [34].

From Table 9, the results on the CEC2020 test suite further highlight the competitiveness of the proposed GWO–ZOA hybrid. In particular, the hybrid achieves the best mean solutions in C20-2–C20-4 and C20-6, outperforming all competitors. These cases demonstrate the algorithm's strong exploitation capability on structured multimodal landscapes as well as its adaptability to hybrid and composition functions. The consistent performance with small standard deviations indicates reliable convergence behavior. A close competitor in this benchmark is PSO, which secures the best performance in C20-1, C20-5, C20-7, C20-8, C20-

Table 10. Statistical results of Wilcoxon rank-sum test of GWO-ZOA

Functions	GWO-ZOA vs PSO (+/=/-)	GWO-ZOA vs SCA (+/=/-)	GWO-ZOA vs MFO (+/=/-)	GWO-ZOA vs HHO (+/=/-)	GWO-ZOA vs GWO (+/=/-)	GWO-ZOA vs ZOA (+/=/-)
23 classical test functions	13/3/7	18/4/1	15/2/6	13/4/6	17/1/5	12/8/3
CEC 2017	10/6/13	29/0/0	23/2/4	19/2/8	6/10/13	27/2/0
CEC 2020	4/0/6	7/1/2	7/0/3	7/1/2	7/0/3	8/1/1
Total	27/9/26	54/5/3	45/4/13	39/7/16	30/11/21	47/11/4

9, and C20-10. These results suggest that, on certain large-scale functions, the simplicity and rapid convergence dynamics of PSO can still provide an advantage over the more complex hybridization.

Despite these individual cases, the Friedman test on the 10 CEC2020 benchmark functions shows a statistically significant performance difference among the algorithms ($\chi^2 = 17.5855$, $p = 0.007356 < 0.05$). The average ranks are: PSO = 2.000, SCA = 5.200, MFO = 4.100, HHO = 3.800, GWO = 3.500, ZOA = 5.200, and GWO–ZOA = 2.700. Since PSO achieved the best overall results, followed by the hybrid algorithm (GWO–ZOA) in second place, this shows that the proposed method performed competitively and was consistently superior to SCA, MFO, HHO, GWO, and ZOA. Notably, SCA and ZOA exhibited the weakest performance, with the highest average ranks (5.200 each). These results confirm that the proposed method is among the strongest performers on the CEC2020 test suite, closely approaching the top-ranked PSO.

3.5. Wilcoxon rank sum test analysis

To further analyze the significance of the experimental results from a statistical perspective, the Wilcoxon rank-sum test was conducted at a significance level of 5% [35, 36]. This nonparametric test is used to compare two groups of paired sample data without assuming normality. For each pair of algorithms, the p-value and h-value were computed to assess whether the differences in performance are statistically significant. In the experiments, a result of p-value < 0.05 and $h = 1$ indicates that the performance distributions of the two algorithms are significantly different [37].

It is crucial to note that the Wilcoxon rank-sum test measures not only the mean values but also the overall distribution of differences. Thus, if the differences are consistent across runs, a significant h-value may be obtained even if an algorithm has a slightly worse mean value on some functions. As a result, the test provides a reliable indicator of statistical significance that goes beyond a simple mean comparison.

Three sets of benchmark functions with different properties, namely the 23 classical benchmark functions, IEEE CEC2017, and IEEE CEC2020 test functions, were subjected to the Wilcoxon rank-sum test. Table 10 provides a summary of the comparison outcomes between the proposed GWO–ZOA algorithm and six other algorithms. According to statistical significance, the symbols $(+/=/-)$ in the table indicate whether GWO–ZOA outperformed, performed equally to, or was worse than the compared algorithm.

The Wilcoxon rank-sum test demonstrates statistical significance ($h = 1$) in specific instances where the mean fitness value of GWO–ZOA is slightly lower. This outcome indicates that the distribution of GWO–ZOA's results is consistently superior or more reliable across multiple iterations, which mean-based comparisons alone may fail to accurately capture. The statistical evidence unequivocally illustrates the robustness and effectiveness of the proposed GWO–ZOA algorithm for solving complex optimization problems.

3.6. Engineering application problems

This section presents two practical engineering examples to demonstrate the efficacy of the GWO–ZOA algorithm in addressing constrained optimization problems with mixed variables. Penalty functions are employed to incorporate the inequality constraints into the objective function.

Prior to the experiments, the parameters of the GWO–ZOA algorithm were set as follows: total population size $N = 190$ and maximum number of iterations $t_{\max} = 800$.

3.6.1. Pressure vessel design problem

The selection of factors in pressure vessel design directly influences the vessel's performance and reliability. Modifying parameters x_1, x_2, x_3, and x_4 will affect the structural integrity, longevity, and pressure resistance of the container. Nonetheless, numerous limitations, specifically g_1, g_2, g_3, and g_4, must be considered when optimizing these parameters to guarantee the vessel's safe functioning under diverse operating situations, with the constraint variables illustrated in Figure 9. This necessitates the incorporation of engineering mathematical models and optimization techniques into the design process to equilibrate parameter interactions and satisfy restrictions.

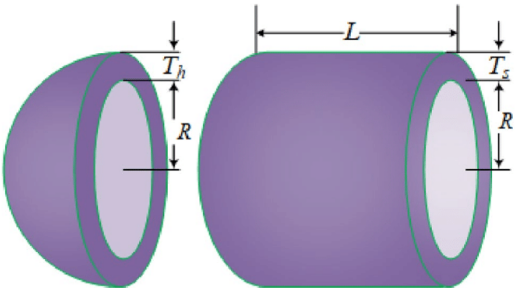


Figure 9. Pressure vessel design problem

The design of pressure vessels aims to reduce manufacturing costs by maximizing essential characteristics within various constraints. The engineering mathematical model is as follows. Consider

$$\mathbf{x} = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L].$$

The objective function is defined as

$$\min f(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3.$$

Subject to the following constraints:

$$\begin{aligned} g_1(\mathbf{x}) &= -x_1 + 0.0193x_3 \le 0, \\ g_2(\mathbf{x}) &= -x_2 + 0.00954x_3 \le 0, \\ g_3(\mathbf{x}) &= -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0, \\ g_4(\mathbf{x}) &= x_4 - 240 \le 0. \end{aligned}$$

The variable ranges are given by

$$0 \le x_1 \le 99, \quad 0 \le x_2 \le 99, \quad 10 \le x_3 \le 200, \quad 10 \le x_4 \le 200.$$

We compare the optimization results of the GWO–ZOA algorithm with the solution results of other algorithms. As can be seen from Table 11, the optimization ability of GWO–ZOA ranks first among these algorithms in dealing with pressure vessel design problems.

Table 11. Comparison of result on pressure vessel design problem

Algorithm	x_1	x_2	x_3	x_4	Fitness
PSO	0.9000	0.4449	46.6320	127.5329	6127.88122
SCA	1.3081	0.6242	65.2548	10.0000	7527.33834
MFO	0.8508	0.4206	44.0833	153.5009	6021.44892
HHO	0.8579	0.4276	44.0606	153.7502	6094.60226
GWO	0.7807	0.3860	40.4373	198.3847	5891.85833
ZOA	0.9859	0.4874	51.0851	89.9632	6343.77792
GWO-ZOA	0.7792	0.3850	40.3497	199.6073	5890.68513

3.6.2. Tension/Compression spring design problem

The tension/compression spring problem is a fundamental structural engineering design problem, whose purpose is to minimize the weight of a tension/compression spring. To solve the problem, three core variables are required: wire diameter (d), mean coil diameter (D), and number of active coils (P). The details of the spring and the three parameters are shown in Figure 10. The mathematical model of this problem is described as follows.

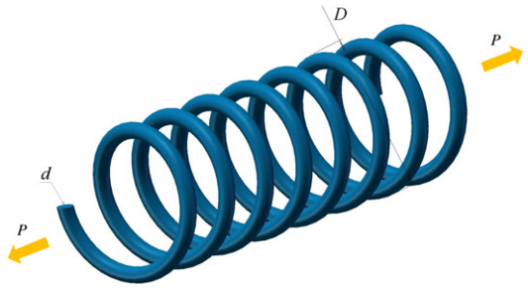


Figure 10. Tension/Compression spring design problem

Consider

$$\mathbf{x} = [x_1, x_2, x_3] = [d, D, P].$$

The objective function is defined as

$$\min f(\mathbf{x}) = (x_3 + 2)x_2x_1^2.$$

Subject to the following constraints:

$$\begin{aligned} g_1(\mathbf{x}) &= 1 - \frac{x_2^2 x_3}{71785x_1^4} \le 0, \\ g_2(\mathbf{x}) &= \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \le 0, \\ g_3(\mathbf{x}) &= 1 - \frac{140.45x_1}{x_2^2 x_3} \le 0, \\ g_4(\mathbf{x}) &= \frac{x_1 + x_2}{1.5} - 1 \le 0. \end{aligned}$$

The variable ranges are given by

$$0.05 \le x_1 \le 2, \quad 0.25 \le x_2 \le 1.30, \quad 2 \le x_3 \le 15.$$

The tension/compression spring problem is optimized, and the values of the relevant parameters are obtained. The optimization results of the GWO–ZOA algorithm are compared with those of other algorithms, and the detailed information is shown in Table 12. The spring weight obtained by the GWO–ZOA algorithm is 0.012675. Overall, the GWO–ZOA algorithm can effectively obtain the optimal solution for engineering problems and determine the best parameter values.

Table 12. Comparison of result on tension/compression spring design problem

Algorithm	x_1	x_2	x_3	Fitness
PSO	0.0530	0.3888	9.6271	0.012695
SCA	0.0599	0.5893	4.6472	0.014069
MFO	0.0500	0.3174	14.0278	0.012719
HHO	0.0579	0.5241	5.5873	0.013313
GWO	0.0500	0.3174	14.0307	0.012721
ZOA	0.0551	0.4446	7.5316	0.012868
GWO-ZOA	0.0511	0.3420	12.2101	0.012675

3.7. Limitations and computational time

Although the proposed GWO–ZOA algorithm demonstrates strong overall performance and robustness, it is not without limitations. The sequential hybridization introduces an additional defense stage, which slightly increases the computation time per iteration compared with the original GWO and ZOA. However, this cost is offset by faster convergence in most test cases, resulting in a comparable or even shorter total runtime to reach the optimal solution.

Furthermore, GWO–ZOA may underperform on very large-scale unimodal problems, where excessive exploitation could reduce diversity prematurely. The algorithm's sensitivity to parameters such as population size and the defense-threshold P_s may also affect its performance consistency across problem domains. Future work should explore adaptive or self-tuning mechanisms to dynamically balance exploration and exploitation and further optimize runtime efficiency.

4. Conclusion

This study proposed a hybrid metaheuristic algorithm, GWO–ZOA, that integrates the exploitation strength of the Grey Wolf Optimizer with the exploration capability of the Zebra Optimization Algorithm. The objective was to achieve a more bal-

anced search process and enhance global optimization performance.

Comprehensive evaluations on 23 classical, 29 CEC2017, and 10 CEC2020 benchmark functions confirmed that the proposed algorithm effectively meets its research objectives. According to the Friedman test, GWO–ZOA achieved the best overall ranking on the classical functions and ranked second on both CEC2017 and CEC2020 suites, closely following PSO. The significant p-values ($p < 0.05$) across all tests indicate that performance differences among algorithms are statistically meaningful. Complementary Wilcoxon rank-sum analysis further verified the consistency of GWO–ZOA, showing a large number of wins and few losses across comparisons, demonstrating its robustness and stable convergence behavior.

The proposed hybrid method also performed successfully on two real-world engineering design problems, outperforming competitors in both feasibility and optimality of solutions. Therefore, the GWO–ZOA algorithm effectively enhances global exploration, convergence accuracy, and overall optimization reliability.

Future work will focus on two key areas: algorithmic refinement and application to complex, high-impact domains. For algorithmic development, efforts will be made to formulate Binary and Multi-Objective variants of GWO–ZOA to address discrete and multi-criteria optimization challenges, respectively. More importantly, the algorithm's strong search balance makes it highly promising for challenging applications. Specific application domains will include feature selection in machine learning, deep learning hyperparameter optimization, and image processing and segmentation, which require the robust global search capabilities inherent in our hybrid approach.

Author Contributions. The sole author, Ayad Ramadhan Ali, conceived, designed, conducted the research, and wrote the manuscript. All aspects of the manuscript, including writing, revising, and final approval, were solely handled by the author.

Acknowledgement. The author would like to express sincere gratitude to the staff of the Department of Mathematics, College of Science, University of Zakho, for their valuable assistance and support. Special appreciation is also extended to the author's family for their continuous encouragement and helpful suggestions throughout the research process.

Funding. The author received no funding for this study.

Conflict of interest. The authors declare no conflict of interest.

Data availability. Data sharing is not applicable to this article as no datasets were generated or analyzed during the study.

References

[1] O. Altay and E. V. Altay, "A novel chaotic transient search optimization algorithm for global optimization, real-world engineering problems and feature selection," *PeerJ Comput. Sci.*, vol. 9, p. e1526, 2023, doi: 10.7717/peerj-cs.1526.

[2] B. F. Azevedo, A. M. A. C. Rocha, and A. I. Pereira, "Hybrid approaches to optimization and machine learning methods: A systematic literature review," *Mach. Learn.*, vol. 113, no. 7, pp. 4055–4097, 2024, doi: 10.1007/s10994-023-06467-x.

[3] S. M. Almufti, A. A. Shaban, Z. A. Ali, R. I. Ali, and J. A. Dela Fuente, "Overview of metaheuristic algorithms," *Polaris Glob. J. Sch. Res. Trends*, vol. 2, no. 2, pp. 10–32, 2023, doi: 10.58429/pgjsrt.v2n2a144.

[4] S. Voß, S. Martello, I. H. Osman, and C. Roucairol, *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Boston, MA: Springer, 2012, doi: 10.1007/978-1-4615-5775-3.

[5] V. C. SS and A. HS, "Nature inspired meta heuristic algorithms for optimization problems," *Computing*, vol. 104, no. 2, pp. 251–269, 2022, doi: 10.1007/s00607-021-00955-5.

[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN'95 Int. Conf. Neural Networks*, IEEE, 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[7] J. H. Holland, "Genetic algorithms," *Sci. Am.*, vol. 267, no. 1, pp. 66–73, 1992, doi: 10.1038/scientificamerican0792-66.

[8] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, pp. 341–359, 1997, doi: 10.1023/A:1008202821328.

[9] X. Yang and A. H. Gandomi, "Bat algorithm: A novel approach for global engineering optimization," *Eng. Comput.*, vol. 29, no. 5, pp. 464–483, 2012, doi: 10.1108/02644401211235834.

[10] S.-E. K. Fateen and A. Bonilla-Petriciolet, "Intelligent firefly algorithm for global optimization," in *Cuckoo Search and Firefly Algorithm: Theory and Applications*, 2014, pp. 315–330, doi: 10.1007/978-3-319-02141-6_15.

[11] X.-S. Yang, *Cuckoo Search and Firefly Algorithm: Theory and Applications*, vol. 516. Cham: Springer, 2013, doi: 10.1007/978-3-319-02141-6.

[12] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, 2019, doi: 10.1007/s00500-018-3102-4.

[13] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[14] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Comput. Struct.*, vol. 169, pp. 1–12, 2016, doi: 10.1016/j.compstruc.2016.03.001.

[15] A. Ali, "A novel approach: Three-group exploration strategy algorithm for solving optimization problems," *Int. J. Inf. Syst. Comput. Sci.*, vol. 9, no. 2, pp. 57–78, 2025, doi: 10.56327/ijiscs.v9i2.1774.

[16] E. Trojovská, M. Dehghani, and P. Trojovský, "Zebra optimization algorithm: A new bio-inspired optimization algorithm," *IEEE Access*, vol. 10, pp. 49445–49473, 2022, doi: 10.1109/ACCESS.2022.3172789.

[17] M. Kohli and S. Arora, "Chaotic grey wolf optimization algorithm for constrained optimization problems," *J. Comput. Des. Eng.*, vol. 5, no. 4, pp. 458–472, 2018, doi: 10.1016/j.jcde.2017.02.005.

[18] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, 2016, doi: 10.1016/j.neucom.2015.06.083.

[19] M. A. Tawhid and A. M. Ibrahim, "A hybridization of grey wolf optimizer and differential evolution for solving nonlinear systems," *Evol. Syst.*, vol. 11, no. 1, pp. 65–87, 2020, doi: 10.1007/s12530-019-09291-8.

[20] W. Lei, W. Jiawei, and M. Zezhou, "Enhancing grey wolf optimizer with Lévy flight for engineering applications," *IEEE Access*, vol. 11, pp. 74865–74897, 2023, doi: 10.1109/ACCESS.2023.3295242.

[21] K. Meidani, A. Hemmasian, S. Mirjalili, and A. Barati Farimani, "Adaptive grey wolf optimizer," *Neural Comput. Appl.*, vol. 34, no. 10, pp. 7711–7731, 2022, doi: 10.1007/s00521-021-06885-9.

[22] X. Zhang and Z. Ming, "An optimized grey wolf optimizer based on a mutation operator and eliminating-reconstructing mechanism and its application," *Front. Inf. Technol. Electron. Eng.*, vol. 18, no. 11, pp. 1705–1719, 2017, doi: 10.1631/FITEE.1601555.

[23] P. Punia, A. Raj, and P. Kumar, "Enhanced zebra optimization algorithm for reliability redundancy allocation and engineering optimization problems," *Cluster Comput.*, vol. 28, no. 4, p. 267, 2025, doi: 10.1007/s10586-024-04931-4.

[24] H. M. El-Hageen *et al.*, "Chaotic zebra optimization algorithm for increasing the lifetime of wireless sensor network," *J. Netw. Syst. Manag.*, vol. 32, no. 4, p. 85, 2024, doi: 10.1007/s10922-024-09860-6.

[25] J. J. Al-Zamili, "Mathematical modeling and optimization of intelligent systems using a hybrid PSO-GWO algorithm: A minx $J(x)$ approach," *Results Nonlinear Anal.*, vol. 8, no. 2, pp. 133–147, 2025, doi: 10.31838/rna/2025.08.02.012.

[26] N. A. Aris *et al.*, "A hybrid chaotic zebra optimization algorithm for cost-effective healthcare team formation," *Int. J. Online Biomed. Eng.*, vol. 21, no. 8, pp. 56–74, 2025, doi: 10.3991/ijoe.v21i08.54697.

[27] X. Yu, Y. Li, and J. Wang, "A hybrid algorithm based on grey wolf optimizer and differential evolution (HGWODE)," *Expert Syst. Appl.*, vol. 213, p. 119327, 2023, doi: 10.1016/j.eswa.2022.119327.

[28] H. M. Mohammed and T. A. Rashid, "A novel hybrid GWO with WOA for global numerical optimization and solving pressure vessel design," *Neural Comput. Appl.*, preprint, doi: 10.36227/techrxiv.11916369.v1.

[29] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, 2016, doi: 10.1016/j.knosys.2015.12.022.

[30] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, 2015, doi: 10.1016/j.knosys.2015.07.006.

[31] A. A. Heidari *et al.*, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, 2019, doi: 10.1016/j.future.2019.02.028.

[32] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[33] G. Wu, R. Mallipeddi, and P. N. Suganthan, *Problem Definitions and Evaluation Criteria for the CEC 2017 Competition on Constrained Real-Parameter Optimization*, Tech. Rep., 2017. [Online]. Available: ResearchGate.

[34] C. T. Yue *et al.*, *Problem Definitions and Evaluation Criteria for the CEC 2020 Special Session and Competition*, Tech. Rep., 2020. [Online]. Available: ResearchGate.

[35] Y. Li, X. Liang, J. Liu, and H. Zhou, "Multi-strategy equilibrium optimizer," *PLoS One*, vol. 17, no. 10, p. e0276210, 2022, doi: 10.1371/journal.pone.0276210.

[36] X. Li, Y. Qi, Q. Xing, and Y. Hu, "IMSCSO: An intensified sand cat swarm optimization with multi-strategy," *IEEE Access*, vol. 11, pp. 122315–122344, 2023, doi: 10.1109/ACCESS.2023.3327732.

[37] Y. Liu, Z. Fang, M. H. Cheung, W. Cai, and J. Huang, "Economics of blockchain storage," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6, doi: 10.1109/ICC40277.2020.9148934.