



OPEN A Geometric Whale Optimization Algorithm with Triangular Flight for Numerical Optimization and Engineering Design

Junhao Wei^{1,2}, Ran Zhang¹, Yanzhao Gu¹, Wenxuan Zhu¹, Yanxiao Li¹, Zikun Li⁴, Shuai Wu^{1,3}, Zhiwen Wang⁵, Ngai Cheong¹, Sio-Kei Im⁶, Yapeng Wang¹ & Xu Yang¹✉

Engineering design optimization problems are often characterized by high dimensionality, complex constraints, and multimodal search landscapes, which pose significant challenges to conventional metaheuristic algorithms. Although the Whale Optimization Algorithm (WOA) has demonstrated competitive performance, it still suffers from premature convergence, limited population diversity, and an imbalanced exploration-exploitation mechanism in complex optimization scenarios. To overcome these limitations, this paper proposes a Geometric Whale Optimization Algorithm (ESTGWOA), in which multiple geometric strategies are systematically embedded into the canonical WOA framework to enhance population initialization, search guidance, and position update behaviors. By incorporating geometric-based mechanisms, ESTGWOA effectively improves search space coverage and strengthens the coordination between global exploration and local exploitation. Comprehensive experiments on 23 benchmark functions demonstrate that ESTGWOA, with an overall effectiveness of 97.10%, outperformed other algorithms on benchmark functions with different dimensions of 30, 50 and 100 dimensions. And the simulations on a series of constrained engineering design problems demonstrate that ESTGWOA consistently outperforms selected state-of-the-art metaheuristic algorithms. Quantitative results show that ESTGWOA achieves superior average fitness values and lower standard deviations in most cases, with statistical significance verified by Wilcoxon rank-sum and Friedman tests. Furthermore, qualitative analyses of search history, population diversity, and convergence behavior confirm the robustness and stability of the proposed approach. These results indicate that ESTGWOA is a reliable and effective optimization algorithm for complex continuous engineering design problems.

Keywords Whale Optimization Algorithm, Triangular flight, Numerical optimization, Engineering design

Metaheuristic algorithms are a class of algorithms used to solve complex optimization problems by employing search and optimization techniques to find the best or near-optimal solutions. Over the past few decades, metaheuristic algorithms have been developed, widely researched, and applied. Due to the complexity and diversity of many real-world problems, traditional exact algorithms often struggle to find optimized solutions within a reasonable time frame. In contrast, metaheuristic algorithms search through the problem space by leveraging the specific structure and domain knowledge of the problem, progressively approaching the optimal solution with minimal computational resources. The core of metaheuristic algorithms lies in Exploration and Exploitation. Exploration refers to thoroughly exploring the entire search space, as the optimal solution could be located anywhere within it. Exploitation involves utilizing effective information as much as possible, as superior solutions often exhibit certain correlations. These correlations are exploited to gradually adjust and guide the search from the initial solution to the optimal solution. In general, metaheuristic algorithms aim to strike a balance between Exploration and Exploitation.

¹Faculty of Applied Sciences, Macao Polytechnic University, Macao 999078, China. ²Pazhou Lab (Huangpu), Guangzhou 510555, China. ³School of Artificial Intelligence, Dongguan City University, Dongguan 523419, China. ⁴School of Economics and Management, South China Normal University, Guangzhou 510006, China. ⁵School of Nursing, Peking University, Beijing 100191, China. ⁶Macao Polytechnic University, Macao 999078, China. ✉email: xuyang@mpu.edu.mo

With the development of computer science, more and more researchers have begun focusing on the development of better metaheuristic algorithms to address complex optimization problems more effectively. As research deepened, metaheuristic algorithms evolved into diverse theoretical and application branches. Based on their underlying inspiration mechanisms, they can generally be categorized into four groups: Evolutionary Algorithms (EAs), Swarm Intelligence Algorithms (SIAs), Physics-Based Algorithms (PBAs), and Human-Based Algorithms (HBAs) in Table 1.

Evolutionary Algorithms are inspired by Darwin's theory of natural selection. Representative examples include the Genetic Algorithm (GA)¹ and Differential Evolution (DE)², which evolve populations through selection, crossover, and mutation operators, demonstrating strong global exploration capability.

Swarm Intelligence Algorithms emulate the social cooperation behaviors of animal groups^{3–6}. For instance, In 1992, Italian scholar Marco Dorigo proposed the Ant Colony Optimization (ACO) algorithm based on the study of ant behavior⁷. In 1995, American psychologists Kennedy and electrical engineer Eberhart, inspired by bird foraging behavior, proposed the Particle Swarm Optimization (PSO) algorithm⁸. In 2014, S Mirjalili et al. proposed the Grey Wolf Optimizer (GWO). GWO simulates the social structure and hunting process of wolf packs⁹. Recently proposed algorithms further expand this category. In 2019, Elhamifar and Ahmadi proposed the Harris Hawks Optimization (HHO) algorithm by mimicking the hunting behavior of Harris hawks¹⁰. In 2021, H Jia et al. inspired by the parasitic behavior of remora, proposed the Remora Optimization Algorithm (ROA)¹¹. In 2022, Eva et al. proposed the Zebra Optimization Algorithm (ZOA), which imitates the foraging behavior of zebras and their defense strategy against predators' attacks¹². In 2024, S Fu et al. proposed Red-billed Blue Magpie Optimizer (RBMO), inspired by the predation and food-storing behaviors of the Red-billed Blue Magpie, which exhibits strong global search capabilities¹³. In 2025, T Hamadneh et al. proposed Salamander Optimization Algorithm (SOA)¹⁴. SOA is designed based on the remarkable biological characteristics of salamanders, including their regenerative abilities, adaptability to diverse environments, and efficient movement strategies. In 2025, X Wang inspired by the social behaviors of bighorn sheep, proposed the Bighorn Sheep Optimization Algorithm (BSOA)¹⁵.

Physics-Based Algorithms are rooted in natural laws of energy transfer and physical dynamics^{16–19}. For example, Simulated Annealing (SA) controls the trade-off between global and local search through a temperature-cooling process²⁰; the Attraction-Repulsion Optimization Algorithm (AROA) models gravitational and repulsive forces²¹; Thunderstorm and Cloud Algorithm (TCA) simulates atmospheric processes observed during thunderstorms²²; Schrödinger Optimizer (SRA) is motivated by principles of quantum mechanics, specifically Schrödinger's equation and wave-particle duality²³.

On the other hand, Human-Based Algorithms are inspired by social learning and cooperative behaviors^{24–27}. The Teaching-Learning-Based Optimization (TLBO) algorithm models classroom teaching and self-learning processes²⁸; the Soccer League Competition Algorithm (SLCA) is based on competitive team dynamics²⁹; Psychologist Algorithm (PA) is inspired by psychological behaviours observed in human psychotherapy³⁰; Perfumer Optimization Algorithm (POA) mimics the behavior of a perfumer when making a perfume³¹.

Metaheuristic algorithms provide a universal optimization strategy, where a single algorithm can be applied to multiple types of optimization problems without requiring a clear mathematical model of the problem. In real-world applications, metaheuristic algorithms have been widely used in various fields due to their strong optimization capabilities, including enhancing wireless sensor network coverage^{32,33}, spring design

Algorithm	Type	Year	Author(s)	Source of Inspiration
Genetic Algorithm (GA)	EAs	1992	Holland et al.	Theory of evolution and Mendelian genetics.
Differential Evolution (DE)	EAs	1995	Storn et al.	Mutation, crossover, and selection mechanisms.
Ant Colony Optimization (ACO)	SIAs	1992	Dorigo et al.	Foraging behavior of ants.
Particle Swarm Optimization (PSO)	SIAs	1995	Kennedy et al.	Social foraging behavior of birds.
Grey Wolf Optimizer (GWO)	SIAs	2014	Mirjalili	Social hierarchy and hunting behavior of grey wolves.
Harris Hawks Optimization (HHO)	SIAs	2019	Heidari et al.	Cooperative hunting strategy of Harris hawks.
Remora Optimization Algorithm (ROA)	SIAs	2021	Jia et al.	Parasitic behavior of remora fish.
Zebra Optimization Algorithm (ZOA)	SIAs	2022	Trojovská et al.	Foraging and defensive behavior of zebras.
Red-billed Blue Magpie Optimizer (RBMO)	SIAs	2024	Fu et al.	Predation and food-storing behavior of red-billed blue magpies.
Salamander Optimization Algorithm (SOA)	SIAs	2025	Hamadneh et al.	Biological characteristics of salamanders.
Bighorn Sheep Optimization Algorithm (BSOA)	SIAs	2025	Wang	Social behaviors of bighorn sheep.
Simulated Annealing (SA)	PBAs	1987	Van Laarhoven et al.	Physical annealing process.
Attraction-Repulsion Optimization Algorithm (AROA)	PBAs	2024	Cymerys et al.	Gravitational attraction and repulsive forces.
Thunderstorm and Cloud Algorithm (TCA)	PBAs	2025	Zraiqat et al.	Atmospheric processes during thunderstorms.
Schrödinger Optimizer (SRA)	PBAs	2025	Hussein et al.	Principles of quantum mechanics.
Teaching-Learning-Based Optimization (TLBO)	HBAs	2011	Rao et al.	Teaching and self-learning processes.
Soccer League Competition Algorithm (SLCA)	HBAs	2014	Moosavian et al.	Competitive dynamics of soccer leagues.
Psychologist Algorithm (PA)	HBAs	2025	Zraiqat et al.	Psychological behaviors and decision-making.
Perfumer Optimization Algorithm (POA)	HBAs	2025	Hamadneh et al.	Behavior of a perfumer in fragrance creation.

Table 1. Current research on standard metaheuristic algorithms.

(stretching/compressing)^{34,35,36}, welding beam design³⁷, neural network tuning^{38–42}, feature selection^{43–45}, urban planning^{46,47}, path planning^{48,49}, antenna design optimization⁵⁰, workshop scheduling⁵¹, and photovoltaic parameter estimation^{52–54}. By efficiently handling complex multi-objective and multi-constrained problems, metaheuristic algorithms significantly enhance decision-making efficiency.

However, metaheuristic algorithms also have certain limitations. These algorithms often face challenges in balancing exploration and exploitation, as well as issues with poor population quality in the later stages of iterations. The Particle Swarm Optimization (PSO) algorithm, when applied to complex optimization problems, is prone to premature convergence, which often leads the algorithm to converge prematurely to a local optimum, halting further exploration for better solutions. The Grey Wolf Optimizer (GWO) is simple in structure and easy to implement, but in complex problems, GWO may also fall into local optima in the later stages of iterations. The Aquila Optimizer, renowned for its strong exploration and exploitation capabilities, faces difficulties due to the complexity of its position update mechanism, making it hard to adjust and unable to guarantee its effectiveness in solving complex problems. Therefore, improving the balance between exploration and exploitation and enhancing population quality have become significant challenges in enhancing the performance of metaheuristic algorithms.

As shown in Table 2, in recent years, researchers have continuously explored integrating different methods to improve metaheuristic algorithms. In 2004, Y. Gao et al. introduced chaotic mapping into population initialization, generating higher-quality populations and improving the optimization ability of the PSO algorithm to some extent⁵⁵. In 2020, Chiwen Qu et al. proposed an improved Harris Hawks Optimization (HHO) algorithm based on information exchange (IEHHO), allowing Harris hawk individuals to exchange information from shared regions, thereby enhancing the collaborative ability of the Harris Hawks to better balance exploration and exploitation⁵⁶. In 2023, H Jia et al. combined heat transfer and condensation strategies, proposing an improved Snow Ablation Optimizer (SAOHTC) to address the issue of premature convergence to local optima in high-dimensional problems⁵⁷. In 2024, J. Huang et al. introduced a multi-strategy hybrid BWO algorithm (HBWO), which integrates QOBL, adaptive spiral predation strategies, and the Nelder-Mead method, to better handle complex high-dimensional problems⁵⁸. These exceptional algorithms, which incorporate various novel improvement strategies, offer new approaches to enhancing metaheuristic algorithms. In 2025, Lu et al. proposed MRBMO by incorporating Good Nodes Set method and LIOBL, for antenna S-parameter optimization⁵⁰.

The Whale Optimization Algorithm (WOA), proposed by Mirjalili et al. in 2016, is a metaheuristic optimization algorithm inspired by the hunting behavior of humpback whales⁵⁹. WOA has a relatively simple structure, making it easy to understand and implement. However, WOA has poor capabilities in balancing exploration and exploitation, and the quality of the population tends to degrade significantly with each iteration, leading to stagnation. In complex multi-modal optimization problems, WOA may fail to perform adequate global exploration and can easily converge prematurely to a local optimum. Therefore, in recent years, many scholars have made various attempts to improve the performance of WOA. In 2020, Liu et al. proposed a hybrid WOA based on Levy flight and Differential Evolution (WOA_LFDE), which used Levy flight to enhance search diversity and combines Differential Evolution (DE) strategies to further improve local search capability while retaining elite individuals⁶⁰. In 2022, S. Chakraborty et al. introduced a novel improved Whale Optimization Algorithm (ImWOA), which incorporated two different exploration strategies for searching food and introduces a new cooperative hunting strategy. ImWOA aimed to address the shortcomings of traditional WOA in solution diversity and the local optimum problem⁶¹. In 2024, Li et al. proposed MISWOA, which incorporated an adaptive nonlinear convergence factor with a variable gain compensation mechanism, adaptive weights, and an advanced spiral convergence strategy, resulting in a significant enhancement in the algorithm's global search capability, convergence velocity, and precision⁶². In 2025, Gu et al. incorporated Good Nodes Set initialization and sine-cosine method into WOA, proposed GWOA⁶³. Their modifications to WOA have improved its applicability to solving more complex problems.

Chapter 2 provided a detailed introduction to the current research on engineering design optimization challenges. Chapter 3 primarily presented the specific principles of the Whale Optimization Algorithm (WOA), along with its advantages and disadvantages. Chapter 4 introduced the proposed ESTGWOA. In Chapter 5, we validated the performance of ESTGWOA through a series of experiments. Chapter 6 involves testing ESTGWOA

Algorithm	Year	Authors	Source of Inspiration
CPSO ⁵⁵	2004	Gao et al.	Chaos mapping for enhancing population diversity.
IEHHO ⁵⁶	2020	Qu et al.	Information exchange mechanism among agents.
WOA_LFDE ⁶⁰	2020	Liu et al.	Levy flight and Differential Evolution operators.
ImWOA ⁶¹	2022	Chakraborty et al.	Dual exploration strategies for global search enhancement.
SAOHTC ⁵⁷	2023	Jia et al.	Heat transfer and condensation mechanisms.
MISWOA ⁶²	2024	Li et al.	Adaptive nonlinear convergence factor and adaptive weights.
HBWO ⁵⁸	2024	Huang et al.	QOBL and adaptive spiral predation strategy.
MRBMO ⁵⁰	2025	Lu et al.	Levy flight and Lens-Imaging Opposition-Based Learning.
GWOA ⁶³	2025	Gu et al.	Good Nodes Set and sine-cosine function guidance

Table 2. Current research on improved metaheuristic algorithms.

with state-of-the-art (SOTA) metaheuristic algorithms on different engineering design optimization problems to verify the practicality and robustness of ESTGWOA.

Current research on engineering design optimization challenges

Engineering design optimization has long been a fundamental task across aerospace, mechanical, civil, and structural engineering. Before the emergence of metaheuristic algorithms, engineering design problems were primarily solved using traditional mathematical optimization techniques such as linear programming, nonlinear programming, dynamic programming, gradient-based methods, and exhaustive search strategies. Although these classical techniques offer strong theoretical foundations, they often struggle with highly nonlinear, multimodal, constrained, and black-box engineering problems. Their reliance on gradient information and strict mathematical assumptions made them unsuitable for complex real-world scenarios, where the design landscapes are typically discontinuous, nonconvex, and computationally expensive.

With the development of metaheuristic algorithms, researchers gained access to a new class of flexible and powerful optimization tools inspired by natural processes, biological systems, and physical phenomena. Algorithms such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), and later more advanced strategies like Grey Wolf Optimizer (GWO) and Whale Optimization Algorithm (WOA) demonstrated strong global search capabilities and robustness. These methods require no gradient information and can handle complex constraints, making them particularly suitable for engineering design optimization involving uncertain environments, noisy models, and intricate design spaces. Metaheuristics quickly became the mainstream approach for tackling challenging engineering design tasks.

In recent years, the integration of neural networks and deep learning has further expanded the landscape of engineering optimization. Surrogate-assisted optimization, reinforcement learning-driven design, and deep surrogate modeling techniques have been widely employed to accelerate computation and improve solution quality. These approaches can approximate computationally expensive engineering models, enabling rapid evaluations and providing new ways to explore high-dimensional design spaces. Despite their effectiveness, neural network-based optimization often requires large datasets, extensive training, and careful hyperparameter tuning, and may still fall short in global search capability compared with well-designed metaheuristics.

Overall, metaheuristic algorithms maintain unique advantages in engineering design, particularly because of their balance between exploration and exploitation, their independence from gradient information, and their ability to escape local optima in complex search landscapes. These strengths have motivated the continuous development of enhanced metaheuristic variants tailored for engineering applications.

The Whale Optimization Algorithm (WOA) is a bio-inspired metaheuristic algorithm known for its simple structure and ease of implementation. However, its performance in engineering design optimization has been less than satisfactory. Therefore, we proposed a novel geometric WOA with multi-strategy (ESTGWOA), aimed at improving the performance of WOA and enabling it to compete with state-of-the-art (SOTA) metaheuristic algorithms, while exploring additional possibilities for WOA in engineering design optimization. This paper will explore the effectiveness and applicability of ESTGWOA in engineering design optimization, aiming to provide a new optimizer for engineering design optimization.

WOA Encircling prey

Humpback whales can identify the location of prey and encircle them. Since the optimal position in the search space is unknown, the WOA assumes that the current best candidate solution represents the target prey or is close to the optimal solution. After defining the best whale, which is called an elite, other whales attempt to update their positions toward the elite. This behavior is represented by Eqs. (1) and 2:

$$D = |C \cdot X^*(t) - X(t)| \quad (1)$$

$$X(t+1) = X^*(t) - A \cdot D \quad (2)$$

where t is the current iteration; A and C are coefficient vectors; X^* is the position of the elite; X is the position of the whale.

If a better solution is found in each iteration, which means the fitness value of X is smaller than that of X^* , the current position X is updated as the new X^* . The formulae for vector A and C are given below:

$$A = 2a \cdot r - a \quad (3)$$

$$C = 2 \cdot r \quad (4)$$

where r is a vector of random numbers from 0 to 1; a is a convergence factor which linearly decreases to 0 during the iteration and the calculation of the convergence factor a is shown in Eq. (5):

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

where t is the current number of iterations; T is the maximum number of iterations.

Fig. 1 illustrates the variation of convergence factor a .

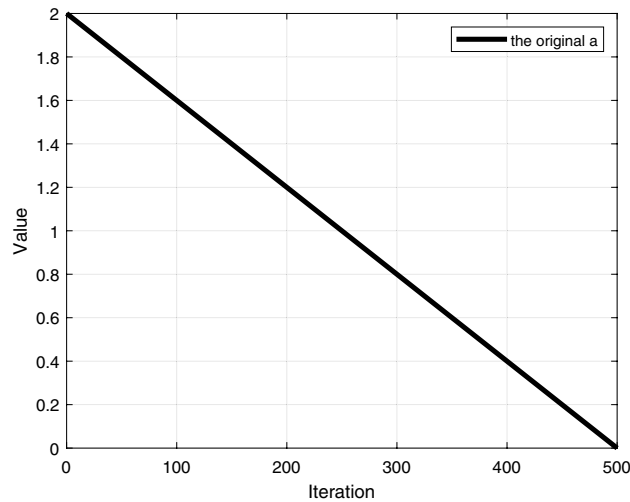


Fig. 1. Variation of convergence factor a during the iteration.

Spiral updating

This method first calculates the whale position X and the prey position X^* in a straight line distance D' from the prey position as shown in Eq. (6). And then creates a spiral equation to simulate the whale spiraling up to encircle the prey as shown in Eq. (7).

$$D' = |X^*(t) - X(t)| \quad (6)$$

$$X(t+1) = D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t) \quad (7)$$

$$a_1 = -1 - \frac{t}{T} \quad (8)$$

$$l = (a_1 - 1) \cdot Rand + 1 \quad (9)$$

where X^* is the position of the elite; and X is the position of the whales; b is a constant defining the shape of the logarithmic helix, usually set to 1; a_1 is a linearly varying parameter of $[-2, -1]$; t is the current number of iterations; T is the maximum number of iterations; $Rand$ is a random number between 0 and 1; the spiral coefficient l takes the values $[-2, 1]$.

Search-for-prey

When the whale moves past the location of the prey, it will abandon its previous direction and randomly search in other directions for other prey, in order to avoid getting trapped in a local optimum. Then a whale individual will be randomly selected from the whale population to explore the area around the position. The modeling of the whale's search-for-prey behavior is as follows:

$$D'' = |C \cdot X_{rand} - X(t)| \quad (10)$$

$$X(t+1) = X_{rand} - A \cdot D'' \quad (11)$$

where C is a coefficient vector calculated in Eq. (4); X_{rand} is a random whale chosen from the current population; A is a coefficient vector calculated in Eq. (3).

Population initialization

Like most metaheuristic algorithms, WOA uses random number initialization for population initialization.

$$X_{i,j} = (ub - lb) \cdot Rand + lb \quad (12)$$

where $X_{i,j}$ is randomly produced population; ub and lb are the upper limit and lower limit of the problem; $Rand$ is a random number between 0 and 1.

This approach, while simple and direct, often results in poor diversity and uneven distribution of solutions, which can lead to inefficiency in the search process.

The pseudo-code of the WOA is shown in Algorithm 1 and the flowchart of WOA is shown in Fig. 2.

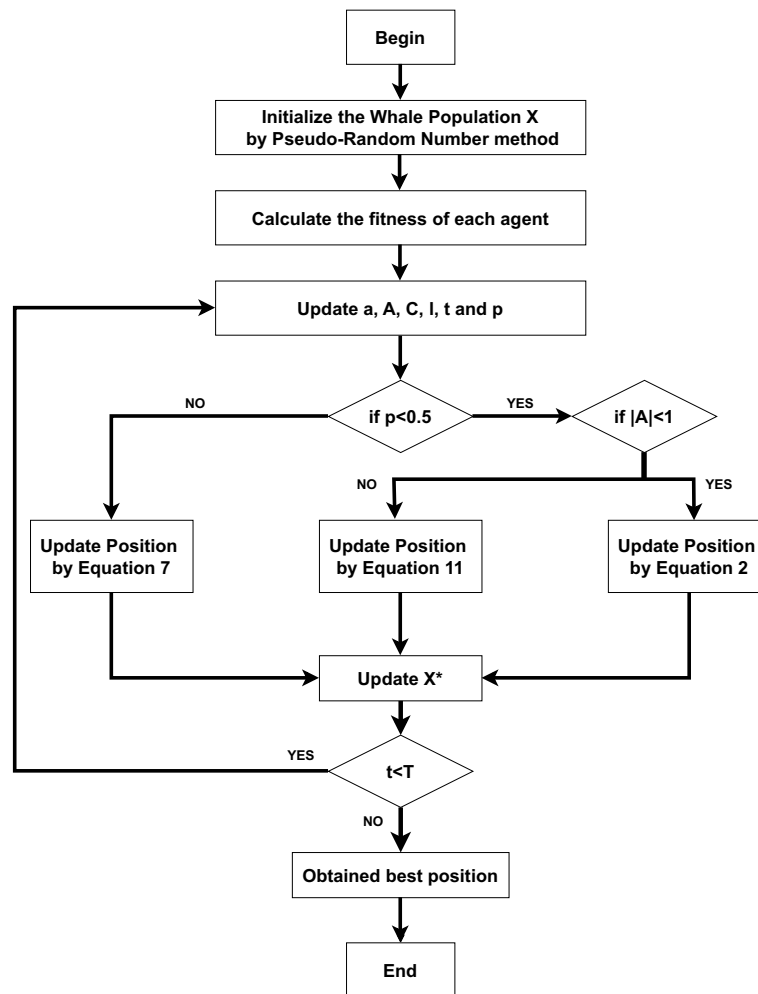


Fig. 2. Flowchart of WOA.

Require: Population size N , Maximum iterations T

Ensure: Best solution X^*

- 1: Initialize the whale population X_i ($i = 1, 2, \dots, N$)
- 2: Calculate fitness of each search agent
- 3: $X^* \leftarrow$ best search agent
- 4: **for** $t = 1$ to T **do**
- 5: $a = 2 - t \cdot \frac{2}{T}$ {Linearly decreases from 2 to 0 as iteration proceeds}
- 6: **for** each search agent **do**
- 7: Update parameters A , C , l , and p
- 8: **if** $p < 0.5$ **then**
- 9: **if** $|A| < 1$ **then**
- 10: $D = |C \cdot X^*(t) - X(t)|$
- 11: $X(t+1) = X^*(t) - A \cdot D$
- 12: **else**
- 13: Select a random search agent X_{rand}
- 14: $D = |C \cdot X_{rand} - X(t)|$
- 15: $X(t+1) = X_{rand} - A \cdot D$
- 16: **end if**
- 17: **else**
- 18: $D' = |X^*(t) - X(t)|$
- 19: $X(t+1) = D' \cdot e^{b \cdot l} \cdot \cos(2\pi l) + X^*(t)$
- 20: **end if**
- 21: **end for**
- 22: Check and correct boundary violations of all search agents
- 23: Calculate fitness of all updated search agents
- 24: Update X^* if a better solution (higher/lower fitness) exists
- 25: $t = t + 1$
- 26: **end for**
- 27: **return** X^*

Algorithm 1. WOA

Advantages and disadvantages of WOA

The structure of WOA is relatively simple, with fewer parameters, making it easy to understand and implement. By dynamically adjusting the convergence factor a , WOA is able to focus on global exploration in the early stages of the search and on local exploitation in the later stages, preventing it from getting trapped in local optima. However, WOA also has certain drawbacks. Despite its ability to balance exploration and exploitation, WOA may not perform sufficient global exploration when dealing with multi-modal optimization problems, leading to premature convergence to local optima, especially in complex search spaces. For some specific problems, WOA's convergence speed is slow, particularly in the later stages when the population diversity significantly decreases, resulting in stagnation. Furthermore, WOA is quite sensitive to the initial setting of parameters, especially the convergence factor a that controls exploration and exploitation. Therefore, there is significant room for improvement in the WOA.

In the Geometric Whale Optimization Algorithm (ESTGWOA) proposed in this paper, we used Good Nodes Set method to generate a uniformly distributed population, introduced a redesigned Elite Guided Search (EGS) strategy to replace the original search-for-prey strategy, and proposed a redesigned Spiral-based Encircling Prey (SEP) strategy to improve the logic of the original encircling prey strategy. We also introduced a Triangular-based Spiral Hunting (TSH) strategy to enhance the original spiral updating strategy, designed a new update mechanism of the convergence factor a to balance the exploration and exploitation, and incorporated a newly-designed Hybrid Gaussian Mutation based on Differential Evolution to help escape from local optima.

ESTGWOA

Good nodes set initialization

The standard WOA initializes its population using pseudo-random numbers, as illustrated on the left side of Fig. 3. Although this strategy is straightforward and easy to implement, it frequently yields populations with limited diversity and uneven spatial coverage. Such deficiencies may cause individuals to cluster, thereby weakening the algorithm's search efficiency and global exploration ability.

To overcome these limitations, this study employs the Good Nodes Set (GNS) initialization method^{64–66}, which generates more uniformly distributed candidate solutions. Originating from the work of Chinese mathematician Loo-keng Hua, GNS provides a systematic way to produce evenly spaced points. Its most notable benefit is that the distribution quality remains consistent regardless of dimensionality, making it suitable for both

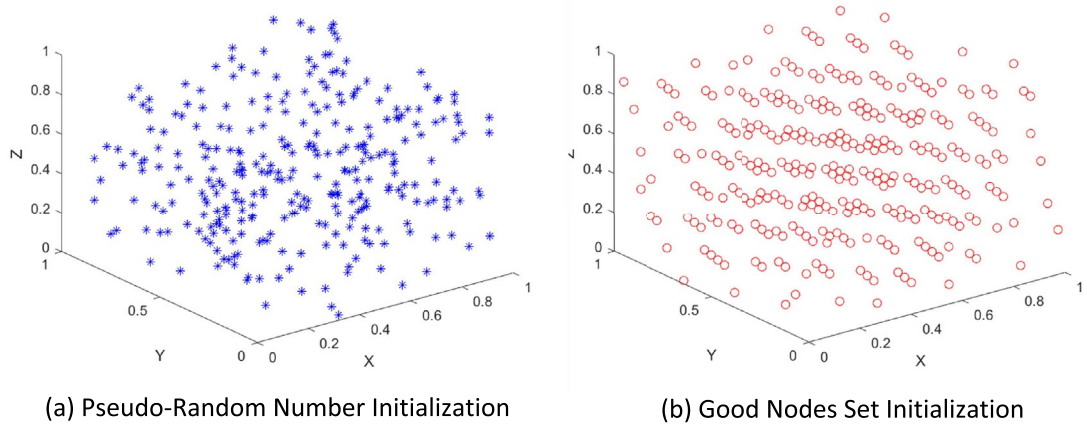


Fig. 3. Pseudo-random number Initialization and Good Nodes Set Initialization.

low-dimensional and high-dimensional optimization tasks. Consequently, applying GNS can enhance the initial population quality and significantly strengthen the exploration phase of WOA. The population generated by GNS for $N=300$ is presented on the right side of Fig. 3, showing a notably more homogeneous layout compared to the pseudo-random counterpart and effectively avoiding concentration of individuals.

Let U^D denote a unit hypercube in a D -dimensional Euclidean space. The construction of a Good Nodes Set is defined by Eq. (13):

$$P_r^M = p(k) = (kr, kr^2, \dots, kr^D) | k = 1, 2, \dots, M \tag{13}$$

where x denotes the fractional part of x , M is the total number of generated nodes, and r is a deviation parameter greater than zero. The constant $C(r, \varepsilon)$ depends solely on r , while ε is a positive constant.

Each element $p(k)$ in P_r^M is referred to as a Good Node. Given the upper and lower bounds of the i^{th} dimension of the search space, x_{max}^i and x_{min}^i , respectively, the mapping from the unit hypercube to the actual search domain is performed using Eq. (14):

$$x_k^i = x_{min}^i + p_i(k) \cdot (x_{max}^i - x_{min}^i) \tag{14}$$

Elite guided searching (EGS) strategy

The original WOA searches for prey by randomly selecting a whale individual, which increases the algorithm’s diversity to some extent. However, this mechanism makes it difficult to achieve an effective balance between global exploration and local exploitation. As a result, the search trajectories of individuals often lack clear directionality and regularity, which may lead to premature convergence to local optima. Moreover, the reliance on randomly selected individuals limits the effective utilization of population information, especially when population diversity decreases in later iterations. This randomness may also cause instability and large performance fluctuations, particularly when the population size is small.

To address these shortcomings, this paper proposes a novel Elite Guided Searching (EGS) strategy, as illustrated in Fig. 4. The EGS strategy is directly inspired by cooperative hunting and predator-prey behaviors observed in the natural world, where individuals dynamically adjust their movements by learning from the most successful hunter (elite) in the group.

The modeling of the Elite Guided Searching strategy is shown below:

$$X(t + 1) = \left(1 - \frac{t}{T}\right) \cdot X^*(t) + |X^*(t) - X_R(t)| \tag{15}$$

$$X_R(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \tag{16}$$

where t is the current iteration number; T is the maximum number of iterations; X represent the positions of the whale individual; X^* represent the position of an elite; X_R represent the average position of all whale individuals.

The Elite Guided Searching strategy uses the position of the current best solution and the average position of the whale individuals to guide the whole population towards the optimal solution. As the iterations progress, the reliance on the optimal solution gradually decreases, allowing the algorithm to naturally balance exploration and exploitation throughout the iteration process. This strategy helps to maintain an appropriate exploration capability in the early stages while enhancing exploitation in the later stages, thus avoiding premature exploitation

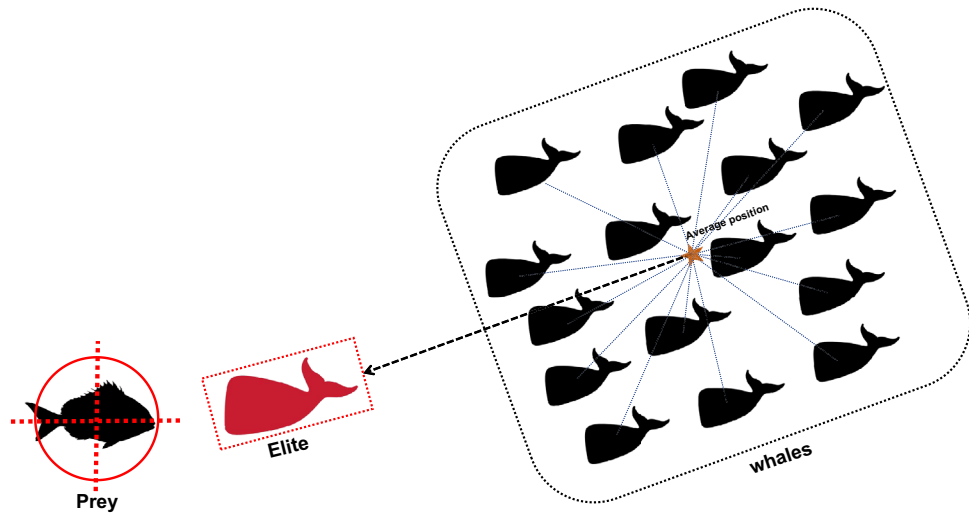


Fig. 4. Elite Guided Searching mechanism.

in the early iterations. Furthermore, the Elite Guided Searching strategy references the average position of the whale population, ensuring that, in the next iteration, the movement direction of a whale individual is influenced not only by the leader’s position but also by the distribution of the entire population. This allows for a more comprehensive use of the population’s information, making the movement of individuals more based on the population’s structural characteristics rather than random behavior. This approach avoids excessive or insufficient interaction among individuals, thereby improving the algorithm’s convergence speed and accuracy. In addition, compared to the original WOA, which explores the unknown domain by randomly selecting a whale individual, the Elite Guided Searching strategy focuses more on the relative positions of the whale individuals and their distance from the current best solution. This reduces the impact of randomness on the algorithm’s performance and gradually guides the whale individuals closer to the optimal solution, enhancing the algorithm’s stability and consistency.

Spiral-based encircling prey (SEP) strategy

From Eq. (2), it can be seen that the encircling prey strategy of the original WOA primarily updates the position of the whale based on the distance between the whale and the prey. Although this approach allows for a gradual convergence towards the optimal solution, in complex and multi-modal problems, it may lead to an overly rapid convergence in the early stages, causing the algorithm to get trapped in local optima. In the original WOA, due to the simplicity of the encircling prey strategy, the position update is solely dependent on the current positions of the prey and the whale, which may lead to insufficient exploration during the local search and make the algorithm sensitive to the initial positions.

To address these limitations, this paper drawn inspiration from the Spiral flight mechanism and introduced a Spiral-based Encircling Prey mechanism. This mechanism is modeled as shown in Eqs. (17) to 23.

Fig. 5 is the simulation of Spiral flight. The schematic of the Spiral-based Encircling Prey is illustrated in Fig. 6.

$$D = |C \cdot X^*(t) - X(t)| \tag{17}$$

$$X(t + 1) = X^*(t) + e^{Z \cdot s} \cdot \cos(2\pi s) \cdot |A \cdot D| \tag{18}$$

$$A = 2a \cdot r - a \tag{19}$$

$$D = |C \cdot X^*(t) - X(t)| \tag{20}$$

$$C = 2 \cdot r \tag{21}$$

$$s = 2 \cdot r - 1 \tag{22}$$

$$Z = e^{S_s \cdot \cos(\pi \cdot (1 - \frac{t}{T}))} \tag{23}$$

where A and C are coefficient vectors; Z and s are the spiral coefficients; $Rand$ denotes a random number between 0 and 1; S_s is the step size of spiral flight.

The Spiral-based Encircling Prey strategy incorporated the concept of spiral flight, introducing periodicity and randomness into the position update process. This spiral motion not only enables the whale to approach the prey but also maintains a certain degree of spatial exploration during the approach, thus preventing premature convergence to local optima. The nonlinear trajectory of this strategy helps the algorithm to escape from local optima, enhancing its global search capability. This strategy introduces a cosine oscillation term, which

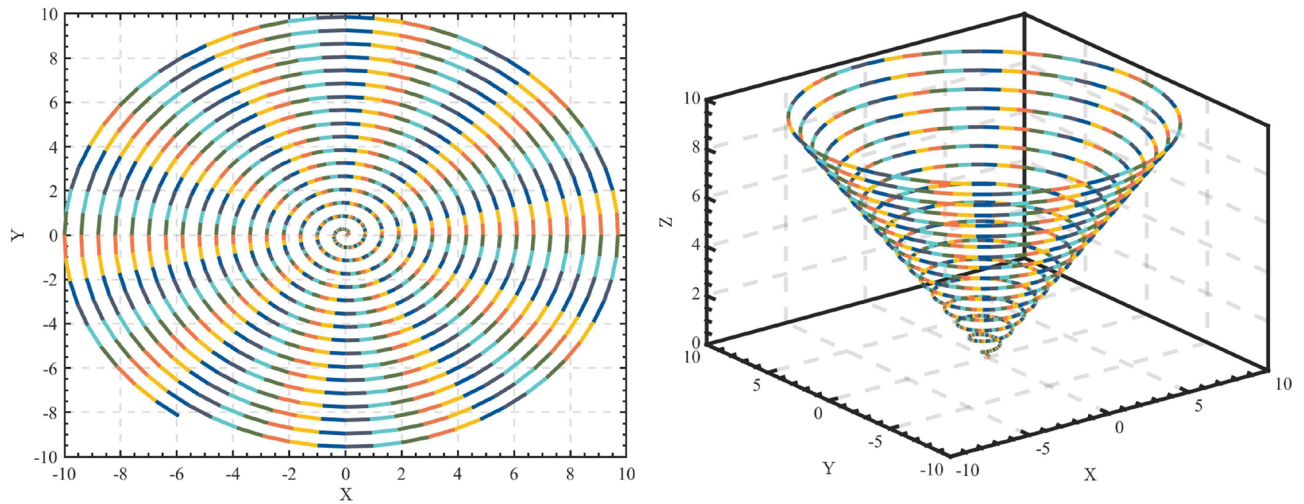


Fig. 5. The simulation of Spiral flight.

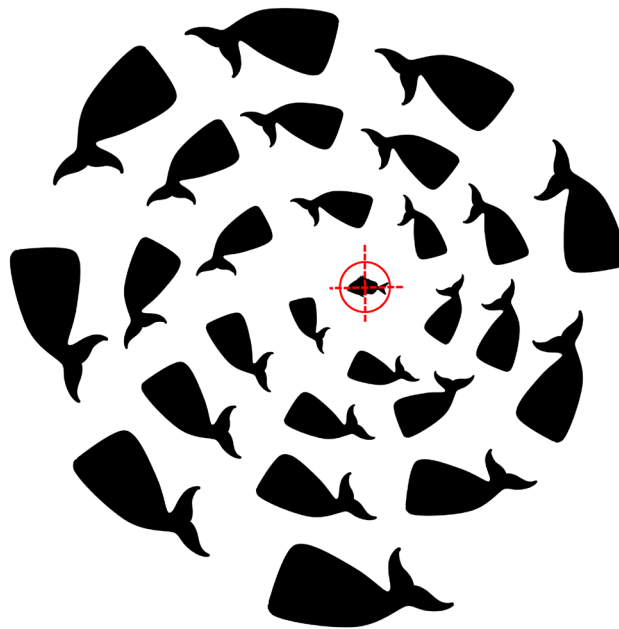


Fig. 6. Spiral-based Encircling Prey mechanism. The whales move in the spiral shape to attack the prey.

increases the depth of exploration of the search space by the whale individual. As a result, the whale can explore the local space more deeply while approaching the prey. This deeper exploration helps in finding better local solutions, preventing early convergence and local optima in complex problems. The movement trajectory of the whale becomes more diversified and complex, which enhances the algorithm's adaptability to various types of problems. This complexity improves the robustness of the algorithm, making it more effective in solving a wide range of optimization problems and mitigating the drawbacks of the original WOA.

Triangular-based spiral hunting (TSH) strategy

From Eq. (7), we can see that the spiral updating strategy of the original WOA primarily relies on position updates around the current best solution, resulting in a high dependency on the current optimal solution. This leads to insufficient fine-tuning ability when approaching the optimal solution. Although this strategy incorporates some randomness, its range is relatively limited, lacking more complex multidimensional space exploration and the ability to escape from local optima, which increases the likelihood of getting trapped in local optima. To address these limitations, this paper proposed a novel Triangular-based Spiral Hunting Strategy to replace the original spiral update strategy, as follows:

$$r_1 = 0.1 \cdot \left(1 - \frac{t}{T}\right) \quad (24)$$

$$r = rand \cdot r_1 \quad (25)$$

$$L_1 = |X^*(t) - X(t)| \quad (26)$$

$$L_2 = D' \cdot rand \quad (27)$$

$$\gamma = 2 \cdot \pi \cdot rand \quad (28)$$

$$L = \sqrt{|L_1 \cdot L_1 + L_2 \cdot L_2 - 2 \cdot L_1 \cdot L_2 \cdot \cos(\gamma)|} \quad (29)$$

$$X(t+1) = X^*(t) \cdot D' + r \cdot L + e^{Z \cdot L} \cdot \cos(2\pi L) \cdot |A \cdot D| \quad (30)$$

where r_1 is the scaling factor; r is the random scaling factor; $rand$ denotes a random number between 0 and 1; L_1 represents the straight-line distance between the whale and the prey; L_2 is the random step size; γ is a random angle; L is the triangular walk step size, as illustrated in Fig. 7.

Compared to the original Spiral updating strategy, the Triangular-based Spiral Hunting Strategy introduces a dynamic scaling random disturbance factor that gradually decreases as the iteration progresses. This guides the particles to explore more widely in the early stages and focus on refined search in the later stages. By calculating the random angle through a cosine function, the strategy introduces more asymmetric disturbances, increasing the uncertainty in particle updates. This prevents all particles from converging onto a single spiral path and further enhances the randomness and directional diversity of the search. The Triangular-based Spiral Hunting strategy allows the population to wander around the optimal position while converging towards it, thereby improving the algorithm's local optimization capability.

Hybrid Gaussian mutation based on differential evolution

The standard position update strategy in WOA primarily relies on biomimetic principles, such as prey encirclement and spiral updating. These strategies typically focus on searching in the vicinity of the current best solution, and especially in the later stages of iterations, WOA may gradually converge to a specific region, leading to a decreased ability to explore the search space. By introducing a mutation strategy after the position update, new disturbances can be introduced into the solution space, creating a certain degree of variation that allows whale individuals to escape from the current local optimum region and explore a broader solution space.

Furthermore, for algorithms like WOA that rely on the best solution in the population for search, a mutation strategy can help individuals escape from locked regions, preventing the algorithm from converging to a local optimum. This is particularly important when solving complex problems with multiple local optima. By incorporating a mutation strategy into WOA, it can compensate for the limitations that may arise during the later stages of convergence, ensuring that WOA continues to explore rather than prematurely converging to a sub-optimal solution. In ESTGWOA, we introduced a novel designed Hybrid Gaussian Mutation based on Differential Evolution.

First, we generate an intermediate solution X''_{new} by Differential Evolution.

$$X''_{new}(t) = X_i(t) + F \cdot ((X_E(t) - X_D(t)) + (X_G(t) - X_F(t))) \quad (31)$$

where X_i represents the current position of the individual; X_D , X_E , X_F and X_G represent the positions of the four different individuals randomly selected from the population respectively; F is the factor controlling the scaling of the difference vector, which is calculated as in Eq. (32).

$$F = 1 + \tan(\pi \cdot (Rand - 0.5)) \quad (32)$$

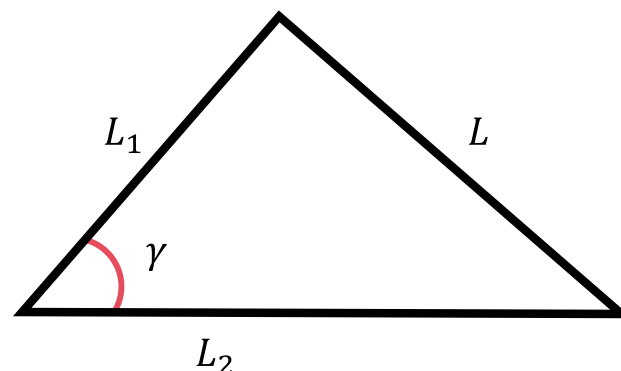


Fig. 7. Step size L of Triangular Walk. L_1 represents the straight-line distance between the whale and the prey; L_2 is the random step size; γ is a random angle; L is the triangular walk step size.

where $Rand$ denotes a random number between 0 and 1.

Then, execute Hybrid Gaussian Mutation on the generated intermediate solutions X'_{new} :

$$X'_{new}(t) = X''_{new}(t) \cdot (1 + \alpha \cdot G_1 + (1 - \alpha) \cdot G_2) \quad (33)$$

where α is the weighting coefficient that controls the mixing ratio of the two Gaussian disturbances; $G_1 \sim N(0, \sigma_1)$ is the first Gaussian-distributed random disturbance with standard deviation; $G_2 \sim N(0, \sigma_2)$ is the second Gaussian-distributed random disturbance with standard deviation. In this study, we set $\alpha=0.5$, $\sigma_1=0.1$ and $\sigma_2=0.5$.

Then, after applying the mutation operation to the intermediate solution, boundary checks and adjustments are necessary to prevent the population from degenerating. If the solution exceeds the upper or lower bounds, it is set to the corresponding boundary value. The boundary checking method is as follows:

$$X_{new}(t) = \min(\max(X'_{new}(t), lb), ub) \quad (34)$$

where ub and lb represent the upper and lower bounds of the problem respectively.

Finally, if the fitness of the mutated solution X_{new} is better than that of the original solution X_i , then use X_{new} to replace X_i .

In the later iteration, WOA typically focuses on utilizing existing solutions for local search to further refine the accuracy of the solution. Introducing this Hybrid Gaussian Mutation based on Differential Evolution can enhance the diversity of the population. By incorporating random disturbances with different scales from Gaussian distributions, it allows the whale population, upon finding a relatively optimal solution, to guide some individuals out of the current region and explore a broader solution space. This improves the algorithm's global search ability and effectively prevents premature convergence to local optima. Such a mutation operation helps the algorithm fine-tune the positions of solutions, increasing the final convergence accuracy and preventing the optimal solution's exact position from being overlooked.

Redesign of convergence factor a

Although the integration of the above strategies improves the performance of WOA, the convergence factor a defined in the traditional WOA no longer meet the specific requirements of the ESTGWOA. The original linear update method of convergence factor a decreases linearly from 2 to 0. This update method results in a gradual change in the algorithm's search behavior throughout the entire iteration process, especially towards the end of the iterations, where the reduction is minimal, affecting the convergence speed. Therefore, this paper proposed a new update mechanism based on the Sigmoid function for the convergence factor a , to balance global exploration and local exploitation. The computation is given by Eq. (35). Fig. 8 compares the proposed Sigmoid-based update with the original linear decrement.

$$a = 2 - \frac{2}{1 + e^{-k(\frac{t}{T} - 0.5)}} \quad (35)$$

where t is the current number of iterations; T is the maximum number of iterations; k is the scaling factor of Sigmoid function.

The Sigmoid function exhibits an 'S-shaped' curve, characterized by slower variation at both ends and rapid change near the midpoint. When applied to parameter updating, this shape produces a gradual decline during

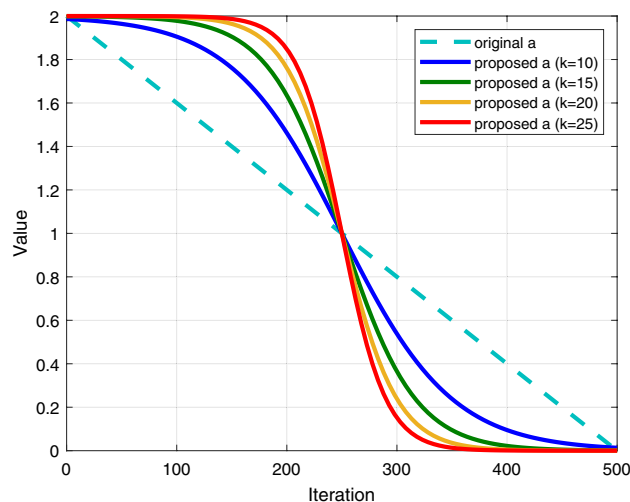


Fig. 8. Comparison between the proposed convergence factor a and the original a . The light blue straight line represents the original convergence factor a ; the blue, green, orange, and red curves represent the proposed convergence factor a for $(k=5)$, $(k=10)$, $(k=15)$ and $(k=20)$, respectively⁶⁶.

the early iterations, a sharp drop in the middle phase, and a slower reduction toward the end. Such a non-linear adjustment pattern effectively models a more sophisticated transition process, allowing the algorithm to display distinct convergence behaviors at different stages. Specifically, the slow early decrease broadens exploration, the mid-phase acceleration enhances convergence speed, and the later mild reduction preserves exploratory potential while strengthening local refinement.

This dynamic adjustment mechanism achieves a more effective balance between global exploration and local exploitation, offering flexible convergence characteristics, mitigating premature convergence, improving robustness across diverse search scenarios, and enhancing the precision of the final solution. The scaling factor k in Eq. (35) governs the evolution of A and therefore directly influences the exploration-exploitation trade-off. The optimal value of k will be determined experimentally in Experiment 5.

The pseudo-code of the complete ESTGWOA is provided in Algorithm 2 and the flowchart of ESTGWOA is shown in Fig. 9.

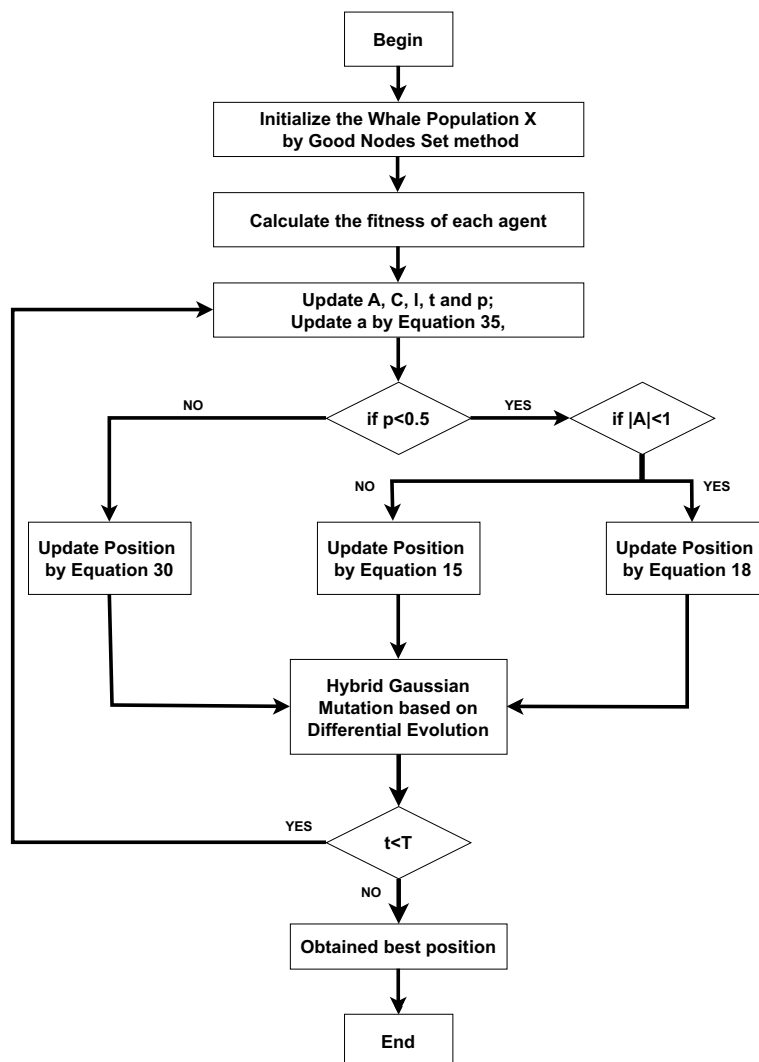


Fig. 9. Flowchart of ESTGWOA.

Require: Population size N , Maximum iterations T

Ensure: Best solution X^*

- 1: Initialize the whale population X_i ($i = 1, 2, \dots, N$)
- 2: Calculate fitness of each search agent
- 3: $X^* \leftarrow$ best search agent
- 4: **for** $t = 1$ to T **do**
- 5: $a = 2 - \frac{2}{1 + e^{-k(\frac{t}{T} - 0.5)}}$
- 6: **for** each search agent **do**
- 7: Update parameters A , C , l , and p
- 8: **if** $p < 0.5$ **then**
- 9: **if** $|A| < 1$ **then**
- 10: $X(t+1) = X^*(t) + e^{Z \cdot s} \cdot \cos(2\pi s) \cdot |A \cdot D|$
- 11: **else**
- 12: Calculate the mean position of the whales $X_R(t) = \frac{1}{N} \sum_{i=1}^N X_i(t)$
- 13: $X(t+1) = (1 - \frac{t}{T}) \cdot X^*(t) + |X^*(t) - X_R(t)|$
- 14: **end if**
- 15: **else**
- 16: $X(t+1) = X^*(t) \cdot D' + r \cdot L + e^{Z \cdot L} \cdot \cos(2\pi L) \cdot |A \cdot D|$
- 17: **end if**
- 18: **end for**
- 19: Execute Hybrid Gaussian Mutation based on Differential Evolution
- 20: Check and correct boundary violations of all search agents
- 21: Calculate fitness of all updated search agents
- 22: Update X^* if a better solution (higher/lower fitness) exists
- 23: $t = t + 1$
- 24: **end for**
- 25: **return** X^*

Algo-

rithm 2. ESTGWOA

Function	Function's Name	Type	Dimension (Dim)	Best Value
F1	Sphere	Uni-modal, Scalable	30 /50/100	0
F2	Schwefel's Problem 2.22	Uni-modal, Scalable	30/50/100	0
F3	Schwefel's Problem 1.2	Uni-modal, Scalable	30/50/100	0
F4	Schwefel's Problem 2.21	Uni-modal, Scalable	30/50/100	0
F5	Generalized Rosenbrock's Function	Uni-modal, Scalable	30/50/100	0
F6	Step Function	Uni-modal, Scalable	30/50/100	0
F7	Quartic Function	Uni-modal, Scalable	30/50/100	0
F8	Generalized Schwefel's Function	Multi-modal, Scalable	30/50/100	-418.98 Dim
F9	Generalized Rastrigin's Function	Multi-modal, Scalable	30/50/100	0
F10	Ackley's Function	Multi-modal, Scalable	30/50/100	0
F11	Generalized Griewank's Function	Multi-modal, Scalable	30/50/100	0
F12	Generalized Penalized Function 1	Multi-modal, Scalable	30/50/100	0
F13	Generalized Penalized Function 2	Multi-modal, Scalable	30/50/100	0
F14	Shekel's Foxholes Function	Multi-modal, Unscalable	2	0.998
F15	Kowalik's Function	Composition, Unscalable	4	0.0003075
F16	Six-Hump Camel-Back Function	Composition, Unscalable	2	-1.0316
F17	Branin Function	Composition, Unscalable	2	0.398
F18	Goldstein-Price Function	Composition, Unscalable	2	3
F19	Hartman's Function 1	Composition, Unscalable	3	-3.8628
F20	Hartman's Function 2	Composition, Unscalable	6	-3.32
F21	Shekel's Function 1	Composition, Unscalable	4	-10.1532
F22	Shekel's Function 2	Composition, Unscalable	4	-10.4029
F23	Shekel's Function 3	Composition, Unscalable	4	-10.5364

Table 3. Classical Benchmark Functions⁶⁷.

Metrics	ESTGWOA ($k=10$)	ESTGWOA($k=15$)	ESTGWOA($k=20$)	ESTGWOA($k=25$)
Average Friedman Value	3.2174	2.5987	2.0765	2.1074
Rank	4	3	1	2

Table 4. Friedman test results of ESTGWOAs with different values of k .

Algorithm	Average Friedman Value	Rank	+/-/-
ESTGWOA1	4.0957	5	7/7/9
ESTGWOA2	4.7884	6	13/7/3
ESTGWOA3	3.7072	4	5/7/11
ESTGWOA4	5.0036	7	14/7/2
ESTGWOA5	3.4094	2	21/0/2
ESTGWOA6	3.6667	3	6/7/10
ESTGWOA	3.3290	1	-

Table 5. Non-parametric results of the algorithms in the ablation study.

Time complexity analysis

Time complexity of WOA

Assume that the time complexity of initialization in WOA is $O(ND)$. During each iteration, the time complexity of boundary checking is $O(ND)$, the time complexity of fitness evaluation is $O(ND)$, and the total time complexity of position updates is $O(ND)$. Therefore, the total time complexity per iteration is $O(ND)$. If the algorithm iterates T times, the total time complexity is calculated as:

$$\text{Total Time Complexity1} = \text{Initialization} + T * (\text{the total time complexity per iteration}) = O(ND) + T * O(ND) = O(T * ND)$$

Time complexity of ESTGWOA

Assume that the time complexity of initialization in ESTGWOA is $O(ND)$. During each iteration, the time complexity of boundary checking is $O(ND)$, the time complexity of fitness evaluation is $O(ND)$, and the total time complexity of position updates is $O(ND)$. Therefore, the total time complexity per iteration is $O(ND)$. If the algorithm iterates T times, the total time complexity is calculated as:

$$\text{Total Time Complexity2} = \text{Initialization} + T * (\text{the total time complexity per iteration}) = O(ND) + T * O(ND) = O(T * ND)$$

In summary, the time complexity of ESTGWOA and WOA are the same, both are $O(T * ND)$.

Experiments and analysis

The experimental environment for this study is as follows: Windows 11 (64-bit), Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz processor, 8GB RAM, and Matlab R2023a as the simulation platform. To verify the performance and effectiveness of ESTGWOA, the following five experiments were designed. The algorithm is tested on a set of 23 selected benchmark functions, as shown in Table 3, and engineering design optimization tests are conducted in Chapter 6.

- *Parameter sensitivity analysis experiment* The four values of the scaling factor k in Eq. (35) were tested on the selected 23 benchmark functions in CEC2005 to determine the optimal value of k that best balances the exploration and exploitation capabilities of the ESTGWOA⁶⁷.
- *Ablation study* Five improved strategies were sequentially removed from the ESTGWOA, and an ablation study was conducted on the benchmark functions;
- *Qualitative analysis experiment* A qualitative analysis experiment was performed by applying ESTGWOA on the 23 benchmark functions to comprehensively evaluate the performance, robustness and exploration-exploitation balance of ESTGWOA in different kinds of optimization problems, by assessing agents' search behavior, exploration-exploitation ratio and population diversity;
- *Comparative experiment* ESTGWOA was compared with other SOTA MAs on the benchmark functions on dimension of 30;
- *Scalability experiment* ESTGWOA was compared with the SOTA MAs on the benchmark functions in higher dimensions on higher dimension of 50 and 100.

Parameter sensitivity analysis experiment

In WOA, the convergence factor a controls the balance between the exploration and exploitation phases of the algorithm. The traditional linear update approach indicates that as the iterations progress, the convergence

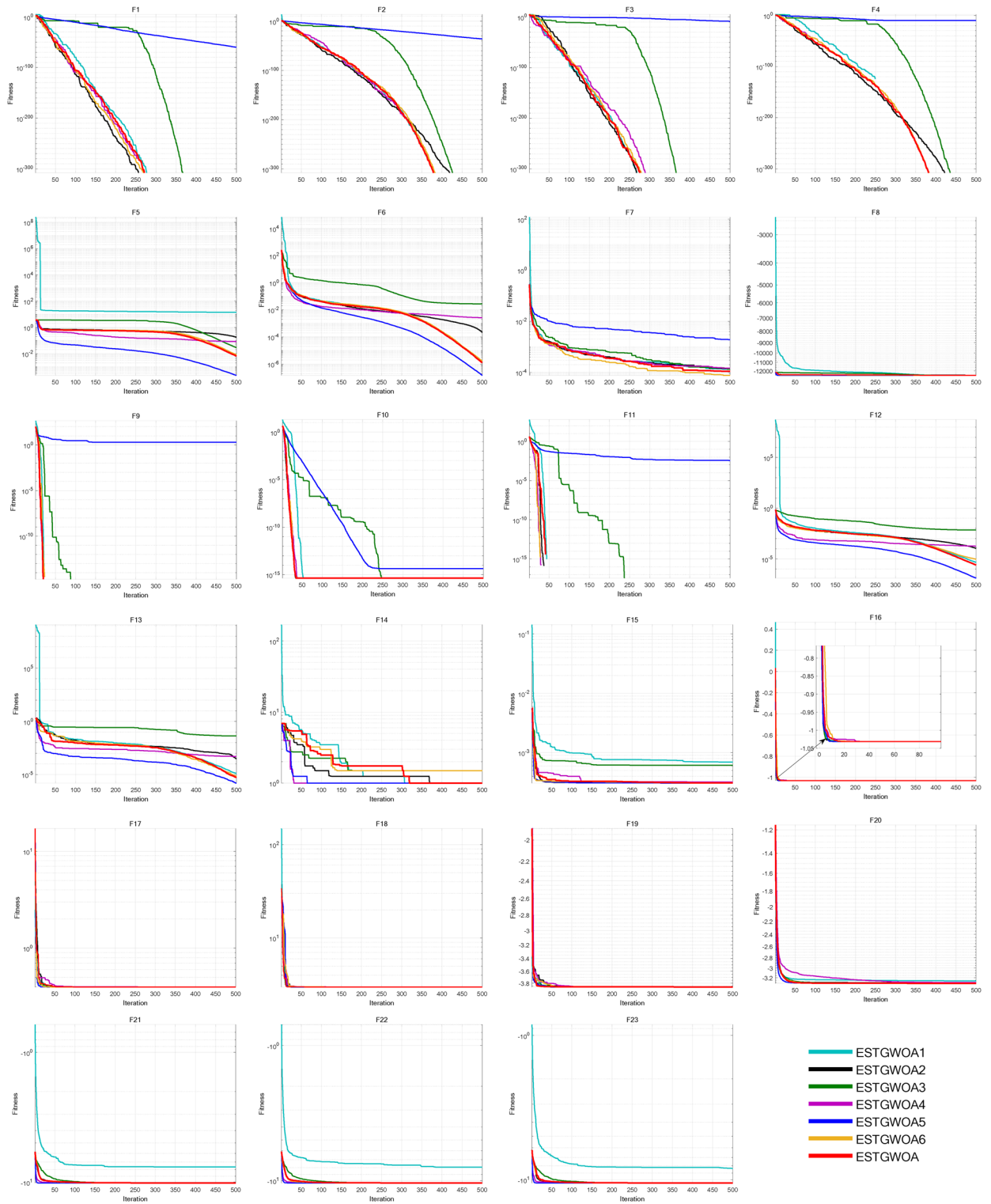


Fig. 10. Iteration curves of the algorithms in ablation study.

factor a linearly decreases from 2 to 0, implying a strong exploration phase in the early iterations and a gradually increasing exploitation phase in the later iterations. However, when a more precise adjustment of the exploration-exploitation balance is required, the linearly varying convergence factor a may limit the adaptability of WOA, making it difficult to balance exploration and exploitation effectively. To address this issue, the update mechanism of convergence factor a based on a Sigmoid function was proposed in this study, replacing the linear update approach and further enhancing WOA's ability to balance exploration and exploitation. However, since

the trend of the Sigmoid-based convergence factor a depends on the value of k , the selection of k is crucial in determining WOA's balance between exploration and exploitation.

The objective of this experiment is to adjust the value of k to influence the trend of convergence factor a , thereby identifying the optimal value of k for the algorithm, enabling a more flexible adjustment of the switch between exploration and exploitation phases. We selected $k=10$, $k=15$, $k=20$, and $k=25$ for comparison, with the corresponding trends of convergence factor a shown in Fig. 8. ESTGWOA with these values of k were tested on 23 benchmark functions using the Friedman test, and the results were presented in Table 4. The results indicated

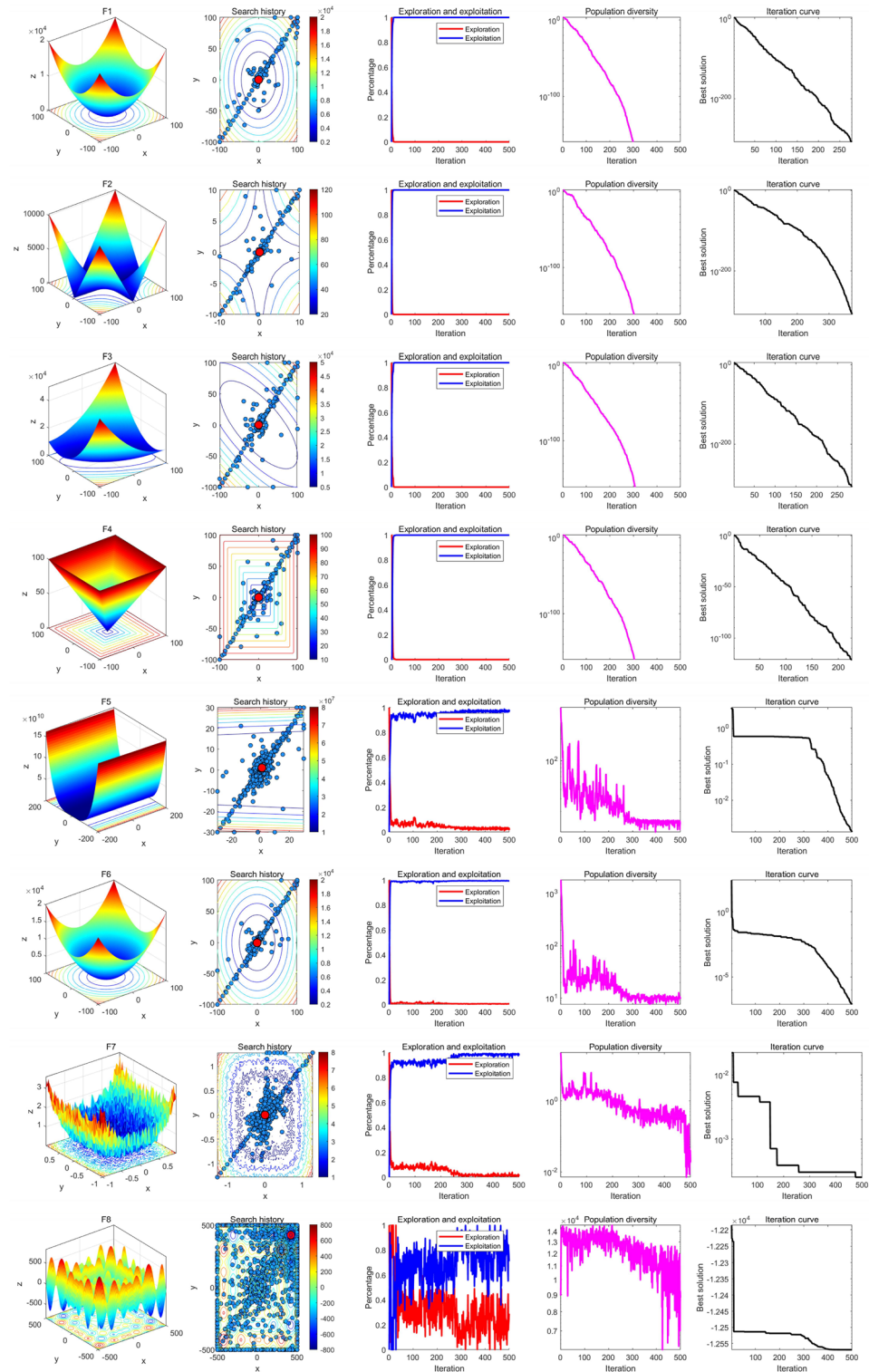


Fig. 11. Results of qualitative analysis experiment (F1-F8).

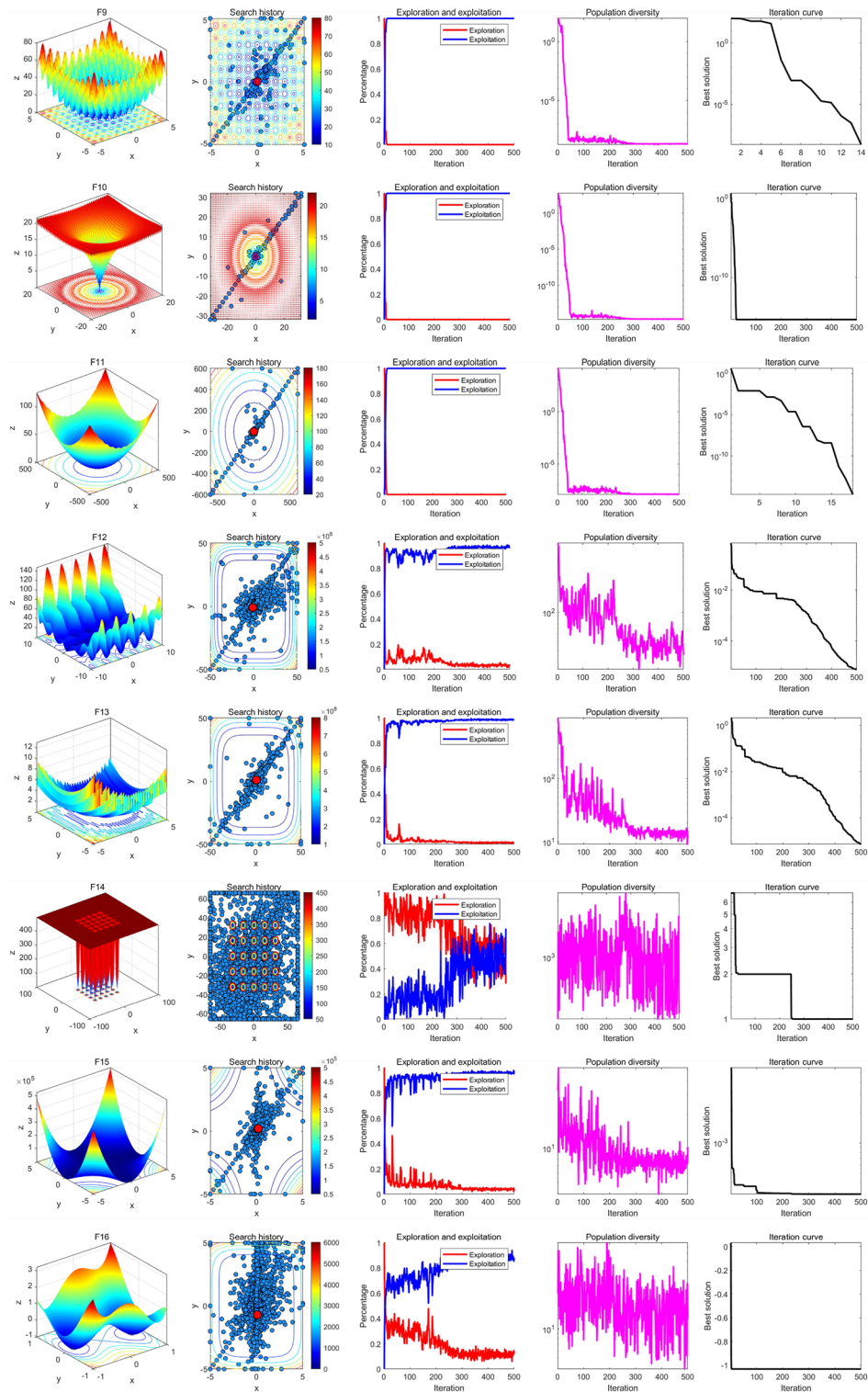


Fig. 12. Results of qualitative analysis experiment (F9-F16).

that, among the 23 benchmark functions, the ESTGWOA with $k=20$ performed the best overall. Therefore, in this study, the scaling factor $k=20$ was chosen for the Sigmoid-based updating mechanism.

Ablation study

In the ablation study, we removed six improvement strategies from ESTGWOA respectively as follows:

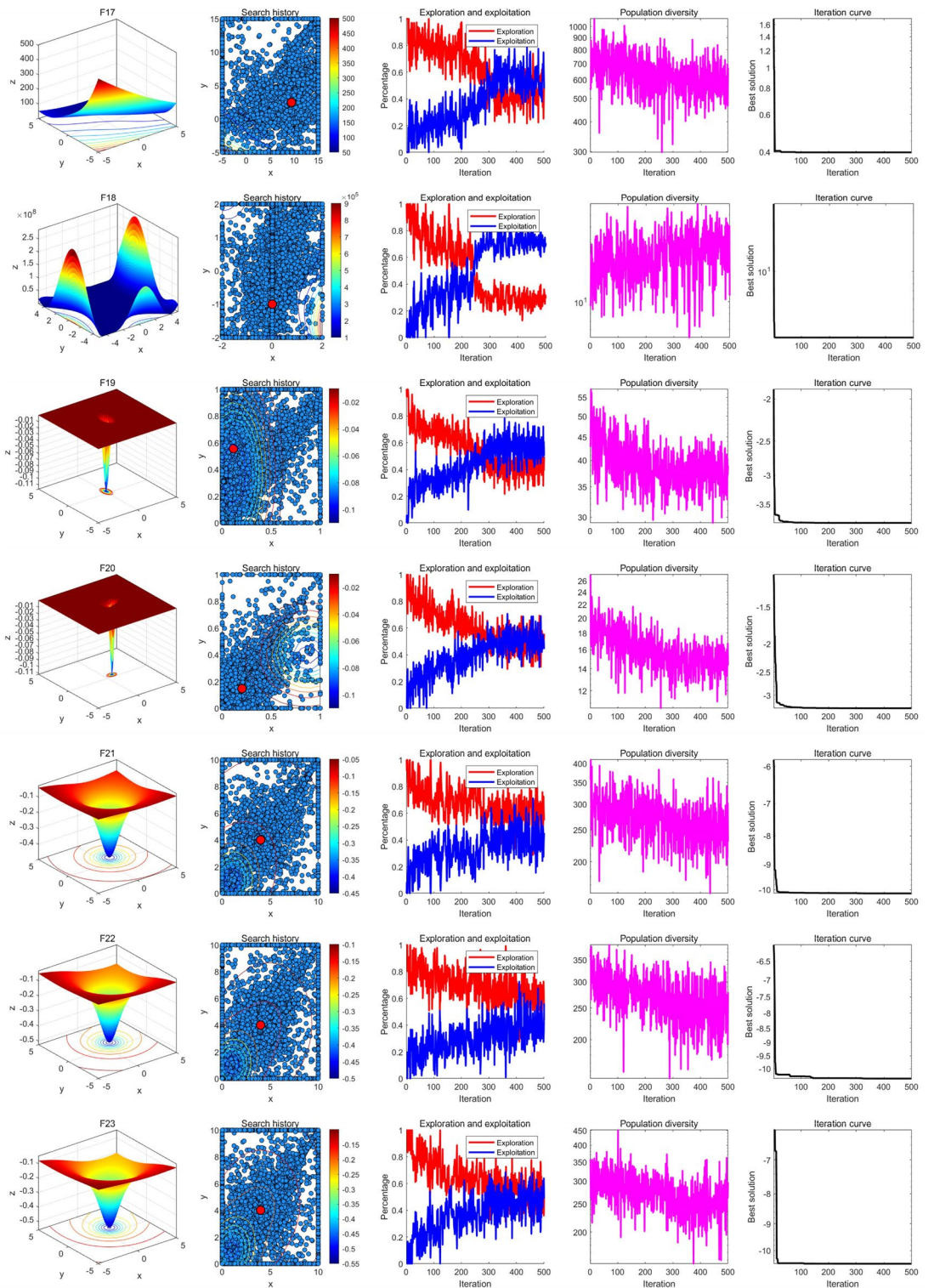


Fig. 13. Results of qualitative analysis experiment (F17-F23).

- ESTGWOA1: The ESTGWOA that Good Nodes Set Initialization was replaced by Pseudo-random number initialization;
- ESTGWOA2: The ESTGWOA that the update method of convergence factor a was replaced by the one in original WOA;
- ESTGWOA3: The ESTGWOA that the Elite Guided Search strategy was replaced with the original WOA search-for-prey strategy;

- ESTGWOA4: The ESTGWOA that the Spiral-based Encircling Prey strategy was replaced with the original WOA encircling prey strategy;
- ESTGWOA5: The ESTGWOA that the Triangular-based Spiral Hunting Strategy was replaced with spiral updating strategy in the original WOA;
- ESTGWOA6: The ESTGWOA without the Hybrid Gaussian Mutation based on Differential Evolution.

Furthermore, the number of iterations was set to $T=500$ and the population size was set to $N=30$ for all experiments. Each algorithm was run 30 times on 23 benchmark functions to perform a performance analysis. Parametric and non-parametric results of the algorithms were recorded in Table 5. The iteration curves were shown in Fig. 10.

From Fig. 10, it could be observed that the Good Nodes Set Initialization produced a more uniformly distributed population, which was advantageous for solving complex multi-modal problems such as F21-F23, significantly improving the algorithm's convergence accuracy and stability. In the Elite Guided Search strategy, as the iterations progress, the movement of the whale individuals gradually decreased their dependence on the optimal solution, allowing the algorithm to more naturally balance exploration and exploitation throughout the iterations. This improved the convergence speed of WOA on F1-F4 and F9-F11 and increases solution accuracy for problems such as F5, F6, F12-F15. The Spiral-based Encircling Prey strategy introduced a certain level of periodicity and randomness through the oscillation term, allowing the algorithm to continually escape local optima. The Triangular-based Spiral Hunting Strategy resulted in faster convergence speed and higher convergence accuracy when solving both simple uni-modal problems such as F1-F4 and complex problems like F9-F11, quickly converging to the optimal value. Additionally, the Sigmoid-based updating method for convergence factor a , as proposed in this study, endowed ESTGWOA with a better balance between global exploration and local exploitation, further enhancing the solution accuracy. This allowed the algorithm to better focus on local exploitation in the later stages of iterations, continuously searching for better solutions. The novel Hybrid Gaussian Mutation based on Differential Evolution increased the diversity of the population, enabling the algorithm to escape from local optima more effectively. As shown in Table 5, the average Friedman value of ESTGWOA is 3.3290, ranking first. This indicates that ESTGWOA is the optimal choice.

Qualitative analysis experiment

In the qualitative analysis experiments, the maximum number of iterations was fixed at $T = 500$, and the population size was set to $N = 30$. Under these settings, ESTGWOA was independently executed on 23 benchmark test functions listed in Table 3 to investigate its search history, exploration and exploitation ratio, and population diversity. To enhance interpretability and comparative analysis, the corresponding landscapes and iteration curves for each test function are also illustrated. The qualitative results are summarized in Fig. 11, Fig. 12, and Fig. 13, which mainly consist of the following aspects:

- Landscapes of the benchmark functions;
- Search history of the whale population;
- Exploration and exploitation ratio curves;
- Population diversity curves;
- Iterative convergence curves of ESTGWOA.

The search history diagram illustrates the spatial distribution of individuals throughout the optimization process. In this visualization, red markers denote the global best solution, whereas blue trajectories represent the movement paths of individual agents. As observed, ESTGWOA is capable of thoroughly covering the search space during the optimization process.

For uni-modal benchmark problems (e.g., F1-F6), ESTGWOA converges rapidly, enabling individuals to locate optimal solutions within a limited number of iterations, which leads to a swift aggregation of agents around the optimum. In contrast, when addressing more challenging landscapes (e.g., F7-F8, F14, and F17-F23) characterized by numerous local optima, ESTGWOA emphasizes extensive global exploration during the initial phases, followed by intensified local refinement in later stages. Consequently, the majority of search trajectories span a broad region of the solution space while gradually clustering near high-quality solutions.

Algorithm	Year	Author(s)	Source of Inspiration
Remora Optimization Algorithm (ROA) ¹¹	2021	Jia et al.	The Remora fish's parasitic behavior in the ocean.
Zebra Optimization Algorithm ZOA ¹²	2022	Trojovská et al.	Foraging and defense strategy of zebras.
Grey Wolf Optimizer (GWO) ⁹	2014	Seyedali Mirjalili et al.	The leadership hierarchy and hunting system of gray wolves.
Attraction-Repulsion Optimization Algorithm (AROA) ²¹	2024	Cymerys	Attraction-repulsion phenomenon.
Harris Hawk Optimization algorithm (HHO) ¹⁰	2019	Heidari et al.	The predatory behavior of Harris's hawks.
MSWOA ⁶⁹	2022	Yang et al.	Adaptive inertia weight ω , dynamic convergence.
MWOA ⁶⁸	2021	Anitha et al.	The cosine function, the correction factors.
IPSO ⁴⁸	2024	Wei et al.	Inertia weight, levy flight and adaptive t -distribution.
Whale Optimization Algorithm (WOA) ⁵⁹	2016	Seyedali Mirjalili et al.	The hunting behavior of humpback whales.

Table 6. Details of the MAs in comparative experiments.

Algorithm	Parameter(s)	Value
ROA ¹¹	C	0.1
	Velocity Factor V	2 decreasing to 0
	Convergence Factor a	-1 decreasing to -2
ZOA ¹²	R	0.1
GWO ⁹	Convergence Factor a	2 decreasing to 0
AROA ²¹	Attraction factor c	0.95
	Local search scaling factor 1	0.15
	Local search scaling factor 2	0.6
	Attraction probability 1	0.2
	Local search probability	0.8
	Expansion factor	0.4
	Local search threshold 1	0.9
	Local search threshold 2	0.85
HHO ¹⁰	Threshold	0.5
	Energy E	$ 2 $ to 0
	Convergence Factor a	2 decreasing to 0
MSWOA ⁶⁹	Spiral Factor b	1
	Inertia Weight ω	From 0.5 to 4.5
	Convergence Factor a	2 decreasing to 0
MWOA ⁶⁸	Spiral Factor b	1
	CF_1	2.5
	CF_2	2.5
	Convergence Factor a	2 decreasing to 0
IPSO ⁴⁸	Inertia Weight ω	0.9 decreasing to 0
	C_1	2
	C_2	2
WOA ⁵⁹	Convergence Factor a	2 decreasing to 0
	Spiral Factor b	1
ESTGWOA	Convergence Factor a	2 decreasing to 0
	k	20
	Spiral Factor b	1
	Step size S_s for Spiral Flight	1
	Spiral Factor s	1

Table 7. Parameter settings for the metaheuristic algorithm in this research.

From the perspective of exploration-exploitation coordination, ESTGWOA exhibits strong adaptive regulation throughout the iterative process. For functions F1-F13, the algorithm maintains a high exploratory tendency at the beginning, then progressively strengthens exploitation, reflecting its robust global optimization capability. Conversely, for functions F17-F23, a stronger exploitation behavior is observed in the early iterations, which then decreases steadily over time, highlighting the algorithm's effectiveness in balancing global and local search strategies.

Moreover, for highly multi-modal functions such as F5-F8 and F12-F23, the population diversity curve of ESTGWOA remains highly fluctuating while sustaining relatively high diversity levels. This behavior indicates that the algorithm effectively prevents premature convergence by avoiding excessive individual aggregation. Overall, ESTGWOA demonstrates superior performance in terms of search coverage, convergence efficiency, and robustness, achieving an effective balance between global exploration and local exploitation.

Comparative experiment

To further assess the performance of ESTGWOA, we conducted comparative experiments involving several well-known MAs, including the Remora Optimization Algorithm (ROA)¹¹, Zebra Optimization Algorithm (ZOA)¹², GWO⁹, Attraction-Repulsion Optimization Algorithm (AROA)²¹, HHO¹⁰, MWOA⁶⁸, MSWOA⁶⁹, IPSO⁴⁸, and WOA. These methods were tested on the benchmark suite listed in Table 3. Algorithm descriptions are provided in Table 6, while their corresponding parameter settings are summarized in Table 7. For consistency, all algorithms were executed with a fixed number of iterations ($T=500$) and a population size of $N=30$. Each approach was independently executed 30 times across the 23 benchmark problems. Performance metrics—including average fitness (Ave), standard deviation (Std), p -values from the Wilcoxon rank-sum test, and Friedman rankings—were collected for comprehensive evaluation. The outcomes of these experiments are presented in Fig. 14, Table 8, and Table 9.

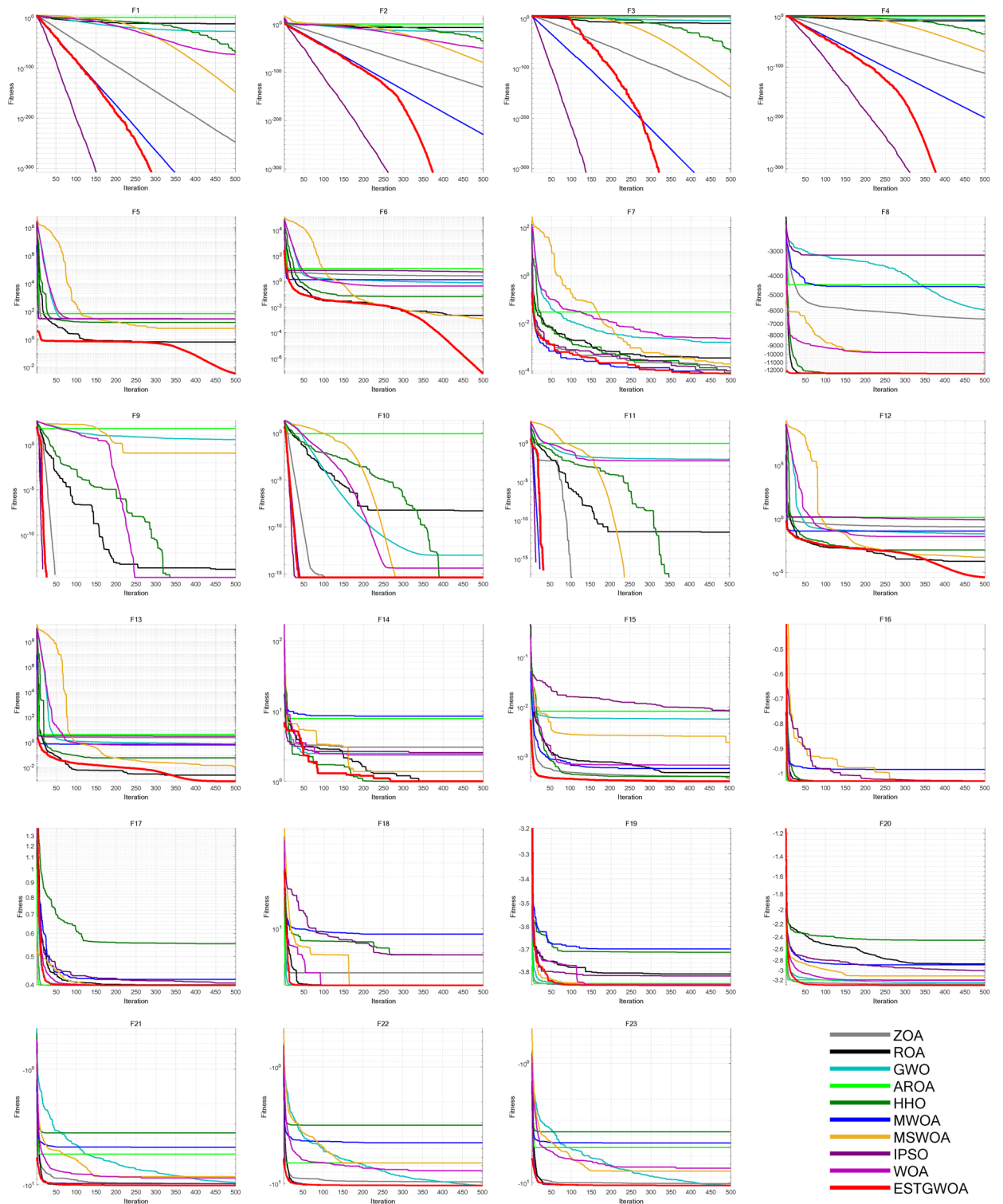


Fig. 14. Iteration curves of different algorithms in comparison experiment.

Parametric analysis

The parametric experimental results demonstrated that ESTGWOA outperformed all other algorithms comprehensively, with a significant improvement in overall performance compared to the original WOA. From Fig. 14 and Table 8, it could be seen that in solving the 23 benchmark functions, ESTGWOA performed the best in terms of both mean and standard deviation across all algorithms. For the majority of functions, ESTGWOA could quickly find the optimal solution, exhibiting fast solution speed and high accuracy, which proves that ESTGWOA had good adaptability and robustness in handling various types of problems.

Function	Metrics	ZOA	ROA	GWO	AROA	HHO	MWOA	MSWOA	IPSO	WOA	ESTGWOA
F1	Ave	4.7827E-249	3.4143E-13	1.0868E-27	4.2403E+00	7.5119E-72	0.0000E+00	1.0998E-148	0.0000E+00	7.9065E-73	0.0000E+00
	Std	5.8640E-249	1.8694E-12	1.5217E-27	3.7448E+00	3.8853E-71	0.0000E+00	4.9824E-148	0.0000E+00	7.9065E-73	0.0000E+00
F2	Ave	1.0595E-131	8.7186E-08	1.0801E-16	7.7261E-01	1.0673E-36	1.5230E-229	7.5956E-81	0.0000E+00	8.6116E-51	0.0000E+00
	Std	4.2324E-131	2.6727E-07	8.9467E-17	2.2061E-01	3.2260E-36	3.2540E-229	1.3748E-80	0.0000E+00	4.2091E-50	0.0000E+00
F3	Ave	2.9921E-159	6.3888E-11	9.0210E-06	2.4139E+02	4.3711E-69	0.0000E+00	3.2595E-138	0.0000E+00	4.6242E+04	0.0000E+00
	Std	1.3928E-158	3.4932E-10	3.0565E-05	2.8941E+02	1.6135E-68	0.0000E+00	5.0813E-138	0.0000E+00	1.3252E+04	0.0000E+00
F4	Ave	7.3762E-113	4.5822E-09	7.6302E-07	1.6511E+00	3.0048E-35	1.8959E-200	1.6591E-70	0.0000E+00	5.5732E+01	0.0000E+00
	Std	3.9055E-112	1.6396E-08	1.0751E-06	6.5084E-01	1.2289E-34	6.7580E-200	1.0996E-70	0.0000E+00	2.4466E+01	0.0000E+00
F5	Ave	2.8311E+01	6.2364E-01	2.6971E+01	6.8446E+01	1.5843E+01	2.8681E+01	5.7916E+00	2.8957E+01	2.7854E+01	3.3540E-03
	Std	4.3081E-01	1.4055E+00	7.8884E-01	3.5585E+01	1.4132E+01	1.0514E-01	1.1623E+01	5.9787E-02	4.8050E-01	3.1925E-03
F6	Ave	2.6897E+00	2.1949E-03	7.7250E-01	9.8772E+00	6.5343E-02	1.3594E+00	1.1669E-03	5.3949E+00	4.2999E-01	6.3342E-08
	Std	6.3008E-01	5.3141E-03	3.9325E-01	2.3398E+00	1.2972E-01	3.9393E-01	1.0052E-03	5.0030E-01	1.9134E-01	2.8733E-08
F7	Ave	1.3892E-04	3.5802E-04	1.5519E-03	2.9988E-02	8.3992E-05	8.1412E-05	1.9592E-04	1.0292E-04	2.3502E-03	7.8903E-05
	Std	1.0183E-04	5.6660E-04	8.7381E-04	2.2690E-02	7.5200E-05	7.3075E-05	1.4485E-04	6.6270E-05	2.1733E-03	2.1623E-05
F8	Ave	-6.6432E+03	-1.2565E+04	-5.9445E+03	-4.4486E+03	-1.2558E+04	-4.5655E+03	-9.8258E+03	-3.1511E+03	-9.8352E+03	-1.2569E+04
	Std	6.7769E+02	2.3363E+01	9.2971E+02	8.5539E+02	6.5075E+01	1.7963E+03	1.7337E+03	9.0291E+02	1.7574E+03	2.9505E-03
F9	Ave	0.0000E+00	1.5158E-14	3.6776E+00	66.1221	0.0000E+00	0.0000E+00	1.1778E-01	0.0000E+00	1.8948E-15	0.0000E+00
	Std	0.0000E+00	8.3025E-14	4.4609E+00	7.4843E+01	0.0000E+00	0.0000E+00	4.4975E-01	0.0000E+00	1.0378E-14	0.0000E+00
F10	Ave	4.4409E-16	5.1079E-09	9.8262E-14	7.7583E-01	4.4409E-16	4.4409E-16	4.4409E-16	4.4409E-16	4.3521E-15	4.4409E-16
	Std	0.0000E+00	2.6780E-08	1.3447E-14	3.2884E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	2.3511E-15	0.0000E+00
F11	Ave	0.0000E+00	2.8039E-12	8.4167E-03	9.7717E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	5.6075E-03	0.0000E+00
	Std	0.0000E+00	1.5323E-11	9.8646E-03	1.4233E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	3.0714E-02	0.0000E+00
F12	Ave	1.8127E-01	1.1265E-04	3.9932E-02	1.3457E+00	1.2661E-03	7.1424E-02	2.6060E-04	8.0409E-01	2.2420E-02	3.7114E-06
	Std	1.0487E-01	2.2805E-04	1.6392E-02	3.5737E-01	3.0240E-03	3.2367E-02	2.7281E-04	2.1450E-01	1.1526E-02	6.4836E-06
F13	Ave	2.2493E+00	2.3699E-03	6.7495E-01	4.1071E+00	5.4806E-02	6.5476E-01	1.1737E-02	2.8845E+00	5.6839E-01	7.5772E-04
	Std	3.6771E-01	3.4465E-03	2.6061E-01	8.9328E-01	9.5711E-02	1.7772E-01	2.0980E-02	1.3686E-01	2.8309E-01	2.8444E-03
F14	Ave	3.0655E+00	9.9800E-01	2.4855E+00	7.8486E+00	9.9800E-01	8.4658E+00	1.3815E+00	2.5621E+00	2.3717E+00	9.9800E-01
	Std	2.9108E+00	1.2126E-10	1.0706E+00	5.2032E+00	3.8971E-07	4.4422E+00	6.5957E-01	3.6463E+00	3.0258E+00	5.0471E-16
F15	Ave	3.7402E-04	4.8204E-04	5.7560E-03	8.3260E-03	4.0450E-04	5.8117E-04	1.9642E-03	8.5771E-03	6.8524E-04	3.2317E-04
	Std	2.0475E-04	3.5087E-04	8.9620E-03	1.0555E-02	1.5511E-04	1.8498E-04	3.9615E-03	1.3517E-02	3.8567E-04	7.3354E-05
F16	Ave	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0311E+00	-1.0316E+00	-9.8447E-01	-1.0310E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00
	Std	9.713E-10	8.1717E-05	1.2772E-08	2.5532E-03	1.407E-06	8.9601E-02	2.4541E-03	1.7161E-03	1.2836E-09	3.9025E-13
F17	Ave	3.9789E-01	3.9854E-01	3.9789E-01	3.9878E-01	3.9798E-01	4.1940E-01	3.9810E-01	4.0228E-01	3.9790E-01	3.9789E-01
	Std	4.2624E-08	1.7219E-03	2.1391E-06	3.7107E-03	2.0556E-04	2.0652E-02	2.6422E-04	5.4481E-03	2.2604E-05	7.1209E-12
F18	Ave	3.9000E+00	3.0016E+00	3.0000E+00	3.0020E+00	5.7025E+00	8.8477E+00	3.0005E+00	5.7124E+00	3.0000E+00	3.0000E+00
	Std	4.9295E+00	5.4157E-03	3.7595E-05	7.3617E-03	8.2442E+00	9.9427E+00	1.6754E-03	1.4829E+01	8.4750E-05	2.0753E-05
F19	Ave	-3.8625E+00	-3.8110E+00	-3.8616E+00	-3.8548E+00	-3.7129E+00	-3.6977E+00	-3.8612E+00	-3.8205E+00	-3.8599E+00	-3.8628E+00
	Std	2.8834E-04	6.5732E-02	2.0999E-03	1.2397E-02	2.3760E-01	2.4344E-01	-3.8612E+00	1.0629E-01	2.8260E-03	2.4915E-05
F20	Ave	-3.2927E+00	-2.8755E+00	-3.2656E+00	-3.1999E+00	-2.4544E+00	-2.8954E+00	-3.1168E+00	-3.0094E+00	-3.2098E+00	-3.3179E+00
	Std	5.3636E-02	1.9346E-01	9.1188E-02	7.6728E-02	6.2403E-01	2.9012E-01	3.2598E-02	2.0562E-01	8.5788E-02	2.2391E-02
F21	Ave	-9.8129E+00	-1.0151E+01	-9.5641E+00	-5.4302E+00	-3.5669E+00	-4.7371E+00	-8.5233E+00	-9.8122E+00	-8.7696E+00	-1.0153E+01
	Std	1.2933E+00	3.2134E-03	1.8285E+00	3.0970E+00	1.8165E+00	7.5983E-01	2.3136E+00	1.6877E+00	2.2957E+00	3.2595E-09
F22	Ave	-9.6941E+00	-1.0396E+01	-1.0401E+01	-6.6719E+00	-3.1577E+00	-4.4668E+00	-6.6748E+00	-1.0363E+01	-7.7806E+00	-1.0403E+01
	Std	1.8377E+00	1.5061E-02	1.0590E-03	3.3229E+00	1.4325E+00	4.3245E-01	3.9909E+00	2.4932E-02	3.0424E+00	2.0166E-09
F23	Ave	-1.0176E+01	-1.0533E+01	-1.0535E+01	-5.0686E+00	-3.7557E+00	-4.6435E+00	-8.0062E+00	-1.0502E+01	-7.5664E+00	-1.0536E+01
	Std	1.3720E+00	5.8658E-03	9.8808E-04	3.3012E+00	1.1318E+00	7.0162E-01	3.4981E+00	2.4035E-02	3.5474E+00	2.6581E-09

Table 8. Results of parametric tests of different algorithms. *Ave* represents average fitness and *Std* represents standard deviation.

In the performance evaluation of algorithms, the average fitness and standard deviation were commonly used to measure the convergence and stability of algorithms. However, they did not intuitively reflect the performance quality of the algorithm. Relying solely on the average fitness and standard deviation for performance comparison has certain limitations. Therefore, non-parametric tests, such as the Wilcoxon rank-sum test and Friedman test, are often introduced. These statistical tests provide deeper analysis and reliability verification.

Algorithm	Average Friedman Value	Rank	+/-/-
ZOA	4.2152	2	20/3/0
ROA	5.3333	4	23/0/0
GWO	5.9406	7	23/0/0
AROA	8.4913	10	23/0/0
HHO	5.5920	5	20/3/0
MWOA	6.4442	9	18/5/0
MSWOA	5.2833	3	21/2/0
IPSO	6.0993	8	16/7/0
WOA	5.8058	6	23/0/0
ESTGWOA	1.7949	1	-

Table 9. Results of non-parametric tests of different algorithms (*Dim*=30).

Dimension	Algorithm	Average Friedman Value	Rank	+/-/-
<i>Dim</i> =50	ZOA	4.2116	2	20/3/0
	ROA	5.2819	4	22/0/1
	GWO	5.8899	7	23/0/0
	AROA	8.5072	10	23/0/0
	HHO	5.5928	5	20/3/0
	MWOA	6.3783	9	18/5/0
	MSWOA	5.1493	3	21/2/0
	IPSO	6.2406	8	16/7/0
	WOA	5.6645	6	22/1/0
	ESTGWOA	2.0841	1	-
<i>Dim</i> =100	ZOA	4.2420	2	19/3/1
	ROA	5.0811	4	22/1/0
	GWO	6.3739	9	23/0/0
	AROA	8.5101	10	23/0/0
	HHO	5.4986	5	20/3/0
	MWOA	6.1812	8	19/4/0
	MSWOA	5.0739	3	21/2/0
	IPSO	6.0971	7	15/7/1
	WOA	5.7522	6	21/2/0
	ESTGWOA	2.1899	1	-

Table 10. Results of non-parametric tests of the algorithms in higher dimensions.

Metrics	ZOA	ROA	GWO	AROA	HHO	MWOA	MSWOA	IPSO	WOA	ESTGWOA
	(w/t/l)	(w/t/l)	(w/t/l)	(w/t/l)	(w/t/l)	(w/t/l)	(w/t/l)	(w/t/l)	(w/t/l)	(w/t/l)
<i>Dim</i> =30	0/3/20	0/0/23	0/0/23	0/0/23	0/3/20	0/5/18	0/2/21	0/7/16	0/0/23	16/7/0
<i>Dim</i> =50	0/3/20	1/0/22	0/0/23	0/0/23	0/3/20	0/5/18	0/2/21	0/7/16	0/2/21	15/7/1
<i>Dim</i> =100	1/3/19	0/1/22	0/0/23	0/0/23	0/3/20	0/4/19	0/2/21	1/7/15	0/2/21	15/7/1
Total	1/9/59	1/1/67	0/0/69	0/0/69	0/9/60	0/14/55	0/6/63	1/21/47	0/4/65	46/21/2
OE	14.49%	2.90%	0.00%	0.00%	13.04%	20.29%	8.70%	31.88%	5.80%	97.10%

Table 11. Effectiveness of ESTGWOA and other algorithms with *Dim*=30, 50 and 100. *w*, *t*, and *l* represent the number of wins, ties, and losses respectively.

Non-parametric analysis

In this section, non-parametric statistical methods are employed to evaluate the significance of performance differences among the compared algorithms. Specifically, the Wilcoxon rank-sum test is used to perform pairwise comparisons between ESTGWOA and each of the baseline methods, assessing whether the observed differences are statistically significant. As shown in Table 9, ESTGWOA had significant differences compared to ROA, GWO and AROA across all benchmark functions. However, ESTGWOA did not show significant differences compared to ZOA, HHO, MWOA, MSWOA and IPSO on F9-F11, as all three algorithms found the optimal value at a

similar speed. Similarly, ESTGWOA did not show significant differences from MWOA on F1 and F3, as both algorithms quickly converged to the optimal value.

Additionally, the Friedman test is conducted to provide a global ranking of all algorithms based on their performance across multiple benchmark functions. These non-parametric approaches are suitable for algorithm evaluation since they do not assume normality in the data distributions. Based on the average Friedman values of each algorithm, Table 9 showed that ESTGWOA had the lowest average Friedman value of 1.7949, ranking first and far surpassing the second-place ZOA (4.2152). MSWOA, ROA, HHO, WOA and GWO had average Friedman values of 5.2833, 5.3333, 5.5920, 5.8058 and 5.9406, ranking third, fourth, fifth, sixth and seventh, respectively. IPSO, MWOA and AROA had average Friedman values of 6.0993, 6.4442 and 8.4913, ranking eighth, ninth and tenth.

In summary, ESTGWOA demonstrated the best overall performance among all algorithms, showing strong competitiveness compared to other excellent MAs.

Scalability experiment

Among the 23 benchmark functions, F1-F13 are scalable in dimensionality, while F14-F23 are limited to fixed dimensions. To assess the adaptability of ESTGWOA to varying problem scales and complexities, the dimensions of F1-F13 were extended to 50 and 100, whereas F14-F23 retained their original dimensionality. Comparative experiments were conducted between ESTGWOA and several other algorithms, including ZOA, ROA, GWO, AROA, HHO, MWOA, MSWOA, IPSO, and WOA, under settings of $Dim=50$ and $Dim=100$. The parameters used for each method are detailed in Table 7, with the iteration count set to $T=500$ and population size to $N=30$. Each algorithm was independently executed 30 times on the test suite. For performance evaluation, the Wilcoxon rank-sum test and Friedman test were applied, and the outcomes are presented in Table 10.

The experimental results indicate that ESTGWOA exhibited superior scalability. In both 50- and 100-dimensional tasks, it ranked first in the Friedman evaluation, as shown in Table 10. Furthermore, the Wilcoxon test revealed statistically significant differences between ESTGWOA and the compared state-of-the-art (SOTA) methods in both dimensional settings. These results confirm ESTGWOA's robust optimization ability and highlight its strong competitiveness among advanced MAs.

A comprehensive summary of performance, measured by the overall effectiveness (OE) metric, is shown in Table 11. Here, w , t , and l represent the number of wins, ties, and losses respectively. OE scores were calculated using Eq. (36) as referenced in⁷⁰.

$$OE = \frac{N - L}{N} \cdot 100 \quad (36)$$

where N is the total number of tests; L is the total number of losing tests for each algorithm.

The results demonstrate that ESTGWOA, with an OE of 97.10%, outperformed other algorithms on benchmark functions with different dimensions of $Dim=30$, $Dim=50$, and $Dim=100$. It proved to be the most effective algorithm among the competitors.

Engineering design optimization challenges

In modern engineering, design optimization plays a critical role in enhancing system performance, reducing costs, and satisfying a wide range of constraints. Engineering design problems are often characterized by nonlinearity, multimodality, strong constraints, and complex variable interactions. Traditional optimization techniques such as gradient-based methods and linear programming often struggle to address these challenges due to limitations like the curse of dimensionality, local optima entrapment, and reliance on gradient or problem-specific information.

MAs have gained increasing attention as robust alternatives, owing to their global search ability, independence from derivative information, and flexibility across diverse problem domains. Algorithms such as PSO⁸, DE² and GWO⁹ have been widely applied to solve complex engineering design problems by mimicking natural phenomena or evolutionary processes.

In order to ensure transparency and reproducibility, the mathematical formulations of all seven engineering design problems considered in this study are explicitly provided in this section, including decision variables, objective functions, and nonlinear constraints. These problems are not only well-established benchmarks in the optimization literature but also originate from real-world engineering contexts such as automotive systems (multi-disk clutch brake), energy infrastructure (gas transmission compressor), chemical processes (reactor network), and structural/mechanical design (I-beam, piston lever, truss, and spring). Their inherent nonlinearity, multimodality, and constraint-coupling reflect the complexity faced in real-world industrial optimization tasks, which are difficult to address effectively with traditional gradient-based or mathematical programming methods. This justifies the adoption of metaheuristic algorithms such as ESTGWOA for robust and scalable optimization.

Despite its popularity due to simplicity and ease of implementation, WOA has shown only moderate performance in many engineering optimization tasks⁵⁹. Its limitations, such as slow convergence and premature stagnation, have restricted its broader applicability in this domain. Therefore, the primary objective of this study is to propose an enhanced variant of WOA-namely, ESTGWOA-to improve optimization performance and further explore the untapped potential of WOA in engineering design optimization.

To rigorously validate the proposed algorithm, this chapter investigates a set of representative engineering design problems, including the multi-disk clutch brake, gas transmission compressor, reactor network, I-beam, piston lever, three-bar truss and tension/compression spring design. These case studies span structural design, mechanical systems, and energy optimization, providing a comprehensive benchmark to assess the robustness and effectiveness of ESTGWOA under constrained and high-complexity conditions. To handle the constraints

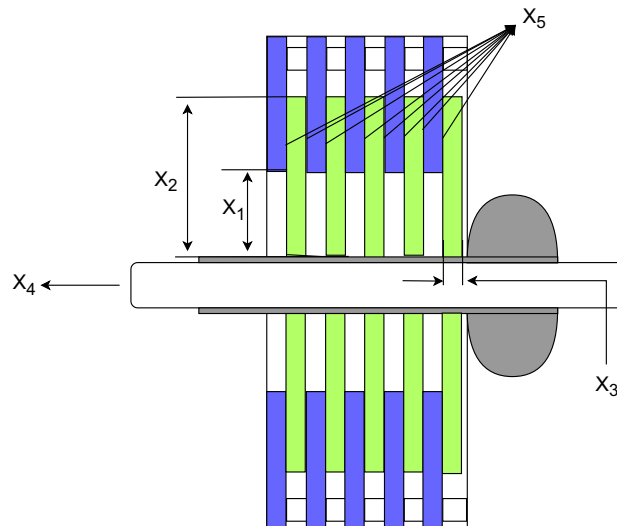


Fig. 15. The structure of a multiple-disc clutch brake.

inherent in these problems, the penalty function method is adopted, transforming constrained optimization problems into unconstrained ones by incorporating constraint violations into the objective function.

All engineering design optimization problems investigated in this study are constrained optimization problems involving nonlinear inequality constraints. To handle these constraints in a unified and implementation-consistent manner, a static penalty function approach is adopted. Specifically, constraint violations are incorporated into the objective function through a quadratic penalty term.

The penalty function is defined as:

$$\text{punishment} = 10^3 \cdot \sum_{i=1}^n \max(0, g_i(x))^2, \quad (37)$$

where n indicates the number of inequality constraints; $g_i(x) \leq 0$ denotes the i^{th} inequality constraint. The penalized fitness function is then formulated as:

$$F(x) = f(x) + \text{punishment}, \quad (38)$$

where $f(x)$ represents the original objective function.

This penalty mechanism assigns increasingly large fitness values to infeasible solutions, thereby discouraging constraint violations and guiding the population toward the feasible region during the optimization process. The same constraint handling strategy is consistently applied to all engineering design problems to ensure fairness and comparability of the optimization results.

Multi-disk clutch brake

The multi-disk clutch brake (MDCB), as shown in Fig. 15, is a critical mechanical component widely used in automotive transmissions and industrial machinery to transmit torque and control rotational motion. The design of an MDCB involves optimizing several geometric and operational parameters to achieve desired performance while minimizing size, weight, or material cost under multiple constraints.

In this study, the MDCB design problem is formulated as a constrained optimization problem with five decision variables:

- X_1 : inner radius of the friction disks;
- X_2 : outer radius;
- X_3 : thickness of the disks;
- X_4 : actuating force applied to engage the clutch;
- X_5 : number of friction surfaces.

The objective is typically to minimize the total mass or cost of the clutch brake system while ensuring sufficient torque transmission capacity and compliance with geometric and physical constraints. Due to the nonlinear and constrained nature of the problem, MAs offer an effective solution approach. The objective function for the MDCB design problem can be described as:

Variable:

$$x = [x_1, x_2, x_3, x_4, x_5]$$

Minimize:

$$y = \pi x_3 \rho (x_2^2 - x_1^2) (x_5 + 1) + \text{punishment} \tag{39}$$

Subject to:

$$g_1(x) = \Delta r + x_1 - x_2 \leq 0; \tag{40}$$

$$g_2(x) = (x_5 + 1)(x_3 + \delta) - l_{\max} \leq 0; \tag{41}$$

$$g_3(x) = P_{rz} - P_{\max} \leq 0; \tag{42}$$

$$g_4(x) = P_{rz} \cdot V_{sr} - P_{\max} \cdot V_{sr\max} \leq 0; \tag{43}$$

$$g_5(x) = V_{sr} - V_{sr\max} \leq 0; \tag{44}$$

$$g_6(x) = T - T_{\max} \leq 0; \tag{45}$$

$$g_7(x) = s \cdot M_s - M_h \leq 0; \tag{46}$$

$$g_8(x) = -T \leq 0; \tag{47}$$

Where:

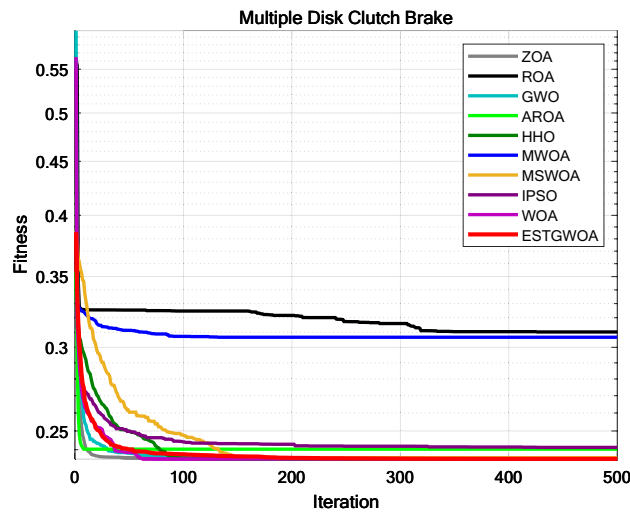


Fig. 16. Iteration curves of the algorithms in MDCB design.

Challenge	Metrics	ZOA	ROA	GWO	ARO	HHO	MWOA	MSWOA	IPSO	WOA	ESTGWOA
		Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
MDCB	Ave	0.235256	0.310220	0.235294	0.240281	0.235243	0.306669	0.235849	0.241194	0.235244	0.235243
	Std	0.000014	0.026709	0.000051	0.010552	0.000003	0.010850	0.000789	0.017186	0.000006	0.000000
GTC	Ave	1224745.938305	1309165.332480	1224745.980931	1226414.350923	1224745.937223	1283139.561224	1224747.548348	1225259.313547	1224745.937231	1224745.937222
	Std	0.003218	16702.572889	0.048508	4318.136982	0.000004	37040.528387	2.044711	302.994825	0.000014	0.000000
RNW	Ave	0.350231	46.011550	0.123568	3.019161	2.181324	47.960433	0.032921	1.647436	1.535390	0.000010
	Std	0.043291	58.353112	0.138384	7.502239	2.662361	55.909959	0.180261	7.222763	1.749428	0.000000
IB	Ave	6.702945	4.526722	6.702792	5.819033	6.702546	5.412604	6.697927	6.701496	6.234270	6.703047
	Std	0.000098	1.720154	0.000226	0.806430	0.000763	1.035780	0.004039	0.001089	0.590366	0.000000
PL	Ave	1.534853	1046.905257	12.164855	242.708852	221.171939	209.122861	6.734300	425.632071	40.159918	1.057186
	Std	0.926179	1714.200946	42.264085	164.447989	161.334757	137.364284	30.814952	642.203111	103.807997	0.000027
TBT	Ave	259.805047	260.312491	259.805064	259.828628	259.817090	260.011082	259.808025	262.342612	259.874216	259.805047
	Std	0.000001	1.627374	0.000020	0.065246	0.013144	0.284944	0.003259	3.211540	0.202731	0.000000
TCS	Ave	0.121523	7.751442	0.121525	0.126589	-	13.670875	0.121546	0.121537	0.121536	0.121522
	Std	0.000001	41.658887	0.000004	0.014248	0.000000	51.534241	0.000028	0.000014	0.000073	0.000000

Table 12. Results of the algorithms in solving engineering design optimization challenges.

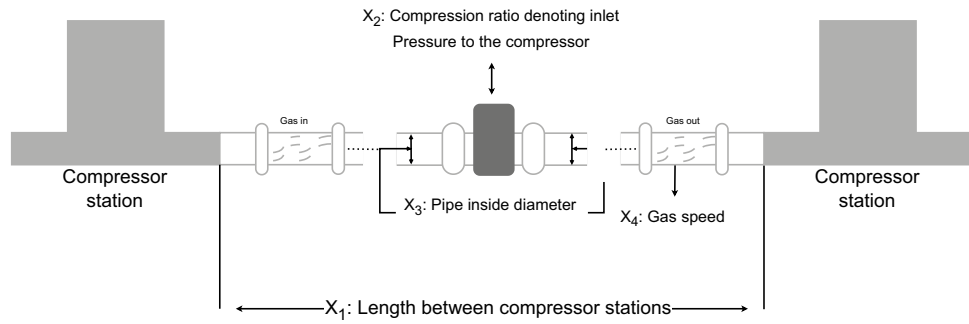


Fig. 17. The structure of a gas transmission compressor system.

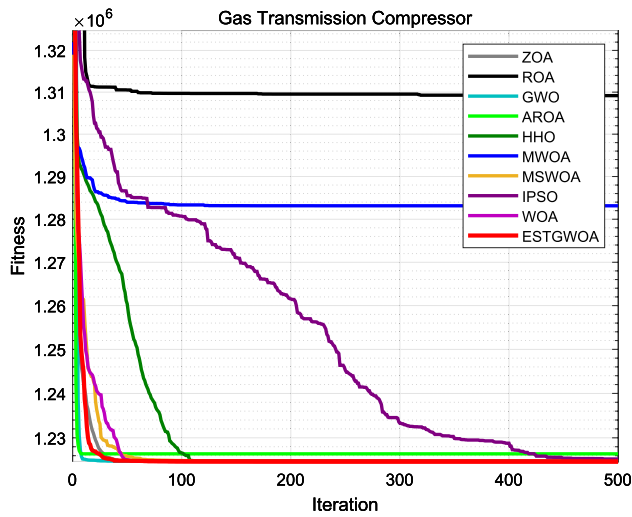


Fig. 18. Iteration curves of the algorithms in GTC design.

$$punishment = 10^3 \cdot \sum_{i=1}^8 \max(0, g_i(x))^2$$

$$M_h = \frac{2}{3} \mu F x_5 \frac{x_2^3 - x_1^3}{x_2^2 - x_1^2};$$

$$P_{rz} = \frac{F}{\pi (x_2^2 - x_1^2)};$$

$$V_{sr} = \frac{2\pi n (x_2^3 - x_1^3)}{90 (x_2^2 - x_1^2)};$$

$$T = \frac{I_z \pi n}{30 (M_h + M_f)};$$

$$P_{rz} = \frac{x_4}{\pi \cdot (x_2^2 - x_1^2)};$$

$$V_{sr} = \pi \cdot R_{sr} \cdot \frac{n}{30};$$

$$R_{sr} = \frac{2}{3} \cdot \frac{x_2^3 - x_1^3}{x_2^2 * x_1^2};$$

$$\Delta r = 20; t_{max} = 3; t_{min} = 1.5; l_{max} = 30; Z_{max} = 10; V_{max} = 10; \mu = 0.6; \delta = 0.5; M_s = 40;$$

$$M_f = 3; n = 250; P_{max} = 1; I_z = 55; T_{max} = 15; F_{max} = 1000; r_{imin} = 55;$$

$$r_{omax} = 110; \rho = 0.0000078;$$

Variable range:

$$60 \leq x_1 \leq 80; \quad 90 \leq x_2 \leq 110; \quad 1 \leq x_3 \leq 3; \quad 0 \leq x_4 \leq 1000; \quad 2 \leq x_5 \leq 9$$

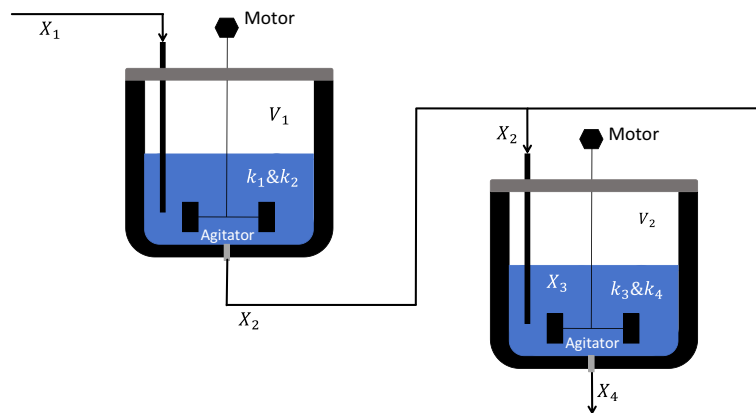


Fig. 19. The structure of a reactor network.

In this study, we conducted comparative tests between ESTGWOA and ZOA, ROA, GWO, AROA, HHO, MWOA, MSWOA, IPSO and WOA. The configuration parameters for all algorithms are detailed in Table 7. The iteration count was uniformly set at $T=500$, with a population size of $N=30$. To ensure reliability, each algorithm was executed independently 30 times on the multi-disk clutch brake design problem. The performance metrics, including *Ave* and *Std*, were recorded for evaluation. The experimental results are shown in Fig. 16 and Table 12. As shown in Table 12, ESTGWOA demonstrated significantly superior optimization accuracy and stability compared to other algorithms. This indicated that ESTGWOA had a substantial advantage in handling such problems.

Gas transmission compressor

The gas transmission compressor (GTC), as shown in Fig. 17, plays a vital role in energy infrastructure, responsible for transporting natural gas from production facilities to end users through a network of pipelines and compressor stations³⁵. The design of an efficient gas transmission system requires a careful balance between construction costs, operational efficiency, and safety constraints, as well as compliance with physical and flow-related limitations.

In this study, the gas transmission system design problem is formulated as a constrained nonlinear optimization task involving four decision variables:

- x_1 : the distance between compressor stations;
- x_2 : the compression ratio, defined as the ratio of inlet to outlet pressure at the compressor;
- x_3 : the inner diameter of the pipeline;
- x_4 : the gas velocity at the output side of the pipeline.

The main objective is to minimize the total annual cost, which typically includes pipeline construction, compressor installation, and operational expenses. The optimization must also satisfy a set of nonlinear constraints related to gas dynamics, pressure loss, pipe capacity, and speed limits. Due to the highly nonlinear and coupled nature of the design parameters, metaheuristic optimization algorithms offer a practical and effective approach for this complex engineering problem. The objective function for the GTC design problem can be described as:

Variable:

$$x = [x_1, x_2, x_3, x_4]$$

Minimize:

$$y = 8.61 \cdot 10^5 x_1^{\frac{1}{2}} x_2 x_3^{-\frac{2}{3}} x_4^{-\frac{1}{2}} + 3.69 \cdot 10^4 x_3 + 7.72 \cdot 10^8 x_1^{-1} x_2^{0.219} - 765.43 \cdot 10^6 x_1^{-1} + \text{punishment} \quad (48)$$

Subject to:

$$g = x_4 x_2^{-2} + x_2^{-2} - 1 \leq 0; \quad (49)$$

Where:

$$\text{punishment} = 10^3 \cdot \max(0, g(x))^2$$

Variable range:

$$20 < x_1 < 50; \quad 1 < x_2 < 10; \quad 20 < x_3 < 45; \quad 0.1 < x_4 < 60$$

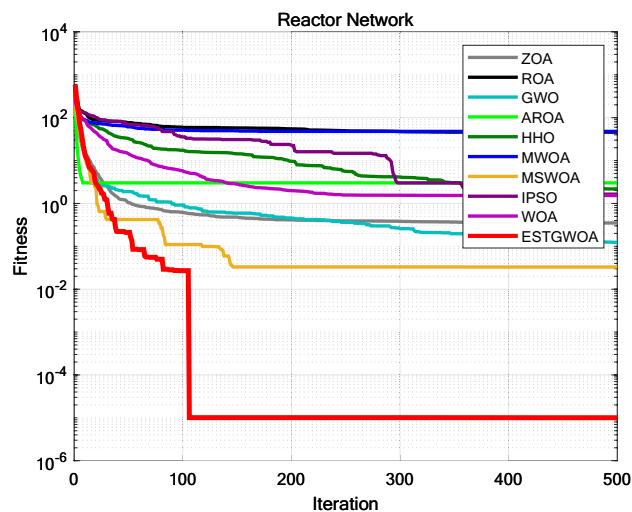


Fig. 20. Iteration curves of the algorithms in RNW design.

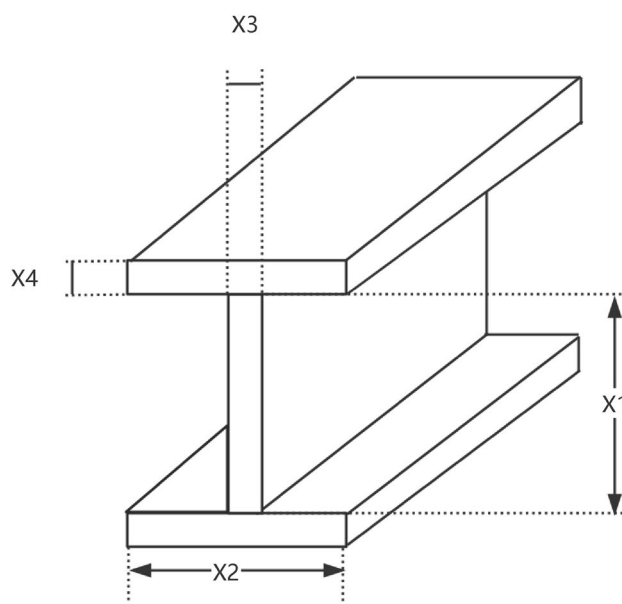


Fig. 21. The structure of an I-beam.

In this study, we conducted comparative tests between ESTGWOA and ZOA, ROA, GWO, AROA, HHO, MWOA, MSWOA, IPSO and WOA. The configuration parameters for all algorithms are detailed in Table 7. The iteration count was uniformly set at $T=500$, with a population size of $N=30$. To ensure reliability, each algorithm was executed independently 30 times on the gas transmission compressor design problem. The performance metrics, including *Ave* and *Std*, were recorded for evaluation. The experimental results are shown in Fig. 18 and Table 12. As shown in Table 12, ESTGWOA demonstrated significantly superior optimization accuracy and stability compared to other algorithms. This indicated that ESTGWOA had a substantial advantage in handling such problems.

Reactor network

The reactor network (RNW) design problem, as shown in Fig. 19, is a critical task in chemical process engineering, aiming to determine the optimal arrangement and operating conditions of chemical reactors to enhance overall process efficiency³⁶. It involves strategic decisions regarding the flow distribution, concentration control, and sequencing of reactors to maximize the final product yield or concentration, while satisfying chemical and physical constraints.

In this study, the reactor network is modeled with four key decision variables, each representing concentrations at various stages of the reaction process:

- x_1 : reactant concentration entering the first reactor;
- x_2 : product concentration leaving the first reactor;
- x_3 : reactant concentration entering the second reactor;
- x_4 : final product concentration at the output.

The optimization focuses on improving reaction performance through fine-tuning reactor configurations and operating parameters, with consideration for reaction equilibrium, mass conservation, and conversion efficiency. Several nonlinear constraints govern the relationships among stages to ensure physical feasibility and chemical consistency. Given the coupled and nonlinear nature of the system, MAs offer a robust approach to solve this class of problems efficiently.

The reactor network optimization problem is subject to several nonlinear constraints to ensure chemical feasibility and system integrity. Specifically, the constraints $h_1(x)$ through $h_4(x)$ are defined as follows:

- $h_1(x)$: ensures the balance of reactants and products within the first reactor, reflecting stoichiometric consistency;
- $h_2(x)$: enforces mass conservation between the outlet of the first reactor and the inlet of the second reactor;
- $h_3(x)$: maintains equilibrium in the reactant concentration levels between the two reactors, accounting for reaction kinetics and system continuity;
- $h_4(x)$: ensures overall mass conservation between intermediate and final products, preserving system closure and yield accuracy;

These constraints are essential for maintaining realistic chemical behavior in the reactor network and preventing physically infeasible or chemically inconsistent configurations. The objective function for the RNW design problem can be described as:

Variable:

$$x = [x_1, x_2, x_3, x_4, x_5, x_6]$$

Minimize:

$$y = x_4 + \text{punishment} \quad (50)$$

Subject to:

$$g(x) = \sqrt{x_5} + \sqrt{x_6} - 4 \leq 0 \quad (51)$$

$$h_1(x) = x_1 + k_1 \cdot x_2 \cdot x_5 - 1 = 0; \quad (52)$$

$$h_2 = x_2 - x_1 + k_2 \cdot x_2 \cdot x_6 = 0; \quad (53)$$

$$h_3(x) = x_3 + x_1 + k_3 \cdot x_3 \cdot x_5 - 1 = 0; \quad (54)$$

$$h_4 = x_4 - x_3 + x_2 - x_1 + k_4 \cdot x_4 \cdot x_6 = 0; \quad (55)$$

Where:

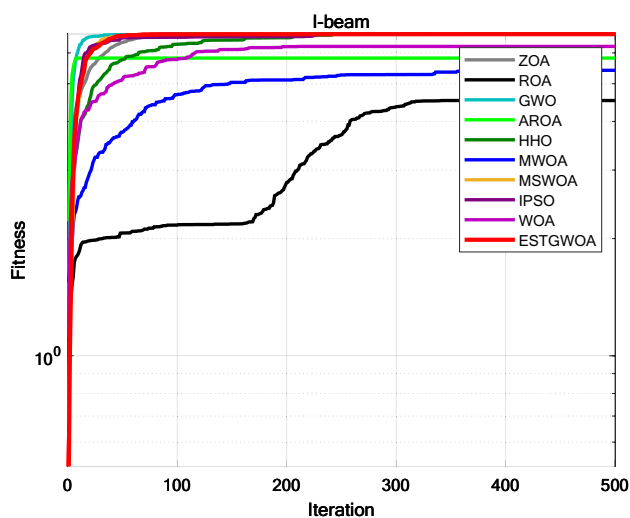


Fig. 22. Iteration curves of the algorithms in I-beam design.

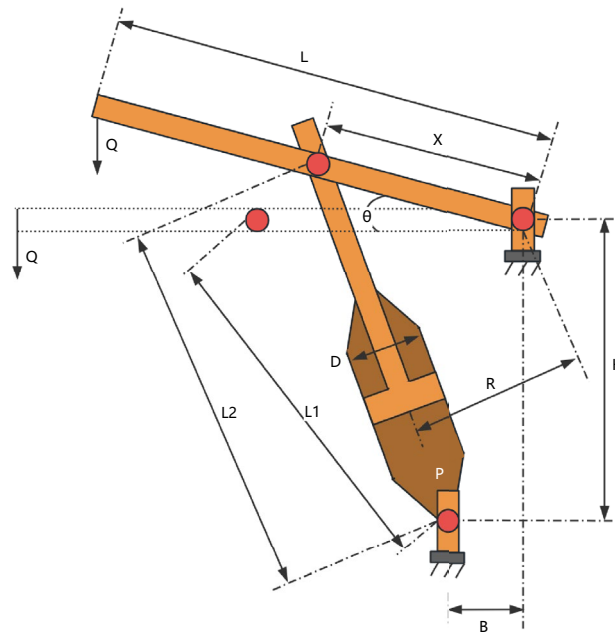


Fig. 23. Structure of a piston lever.

$$k_1 = 0.09755988; \quad k_2 = 0.99 \cdot k_1; \quad k_3 = 0.0391908; \quad k_4 = 0.9 \cdot k_3;$$

$$punishment = 10^3 \cdot \max(0, g(x))^2 + 10^3 \cdot \sum_{i=1}^4 h_i(x)^2$$

Variable range:

$$0.00001 \leq x_1 \leq 1; \quad 0.00001 \leq x_2 \leq 1; \quad 0.00001 \leq x_3 \leq 1; \quad 0.00001 \leq x_4 \leq 1;$$

$$0.00001 \leq x_5 \leq 16; \quad 0.00001 \leq x_6 \leq 16;$$

In this study, we conducted comparative tests between ESTGWOA and ZOA, ROA, GWO, AROA, HHO, MWOA, MSWOA, IPSO and WOA. The configuration parameters for all algorithms are detailed in Table 7. The iteration count was uniformly set at $T=500$, with a population size of $N=30$. To ensure reliability, each algorithm was executed independently 30 times on the reactor network design problem. The performance metrics, including *Ave* and *Std*, were recorded for evaluation. The experimental results are shown in Fig. 20 and Table 12. As shown in Table 12, ESTGWOA demonstrated significantly superior optimization accuracy and stability compared to other algorithms. This indicated that ESTGWOA had a substantial advantage in handling such problems.

I-beam

The I-beam (IB) design optimization problem, as shown in Fig. 21, is a constrained structural design task aimed at minimizing material usage while ensuring mechanical strength³⁵. It involves four decision variables: the web height (x_1), flange width (x_2), web thickness (x_3), and flange thickness (x_4). Two nonlinear constraints, $g_1(x)$ and $g_2(x)$, are imposed to ensure the design meets stress and deflection requirements under loading conditions. The problem reflects typical scenarios in civil and mechanical engineering where lightweight and strength-efficient designs are essential.

The objective function for the I-beam design problem can be described as:

Variable:

$$x = [x_1, x_2, x_3, x_4]$$

Maximize:

$$y = \frac{5000}{\frac{x_3(x_1 - 2x_4)^3}{12} + \frac{x_2 \cdot x_4^3}{6} + 2x_2 \cdot x_4 \left(\frac{x_1 - x_4}{2}\right)^2} + punishment$$

Subject to:

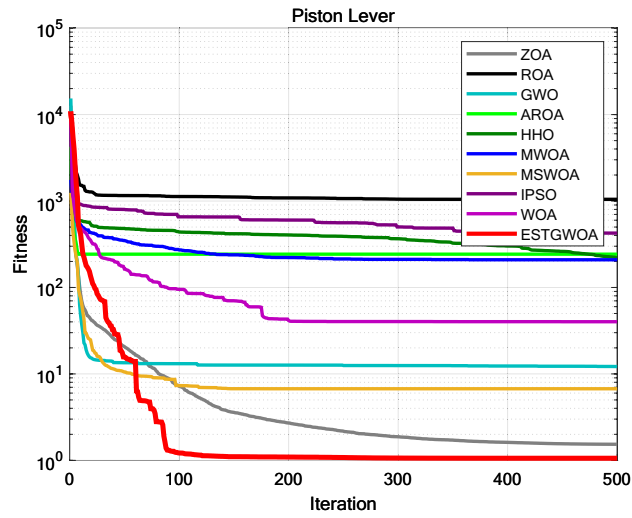


Fig. 24. Iteration curves of the algorithms in Piston Lever design.

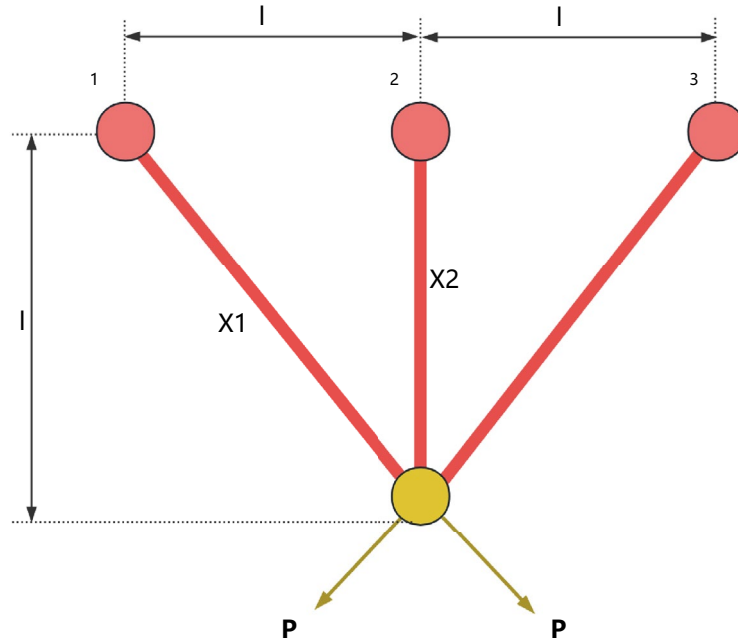


Fig. 25. The structure of a three-bar truss.

$$g_1(x) = 2x_2 \cdot x_3 + x_3 \cdot (x_1 - 2x_4) - 300 \leq 0;$$

$$g_2(x) = \frac{18 \cdot 10^4 x_1}{x_3 (x_1 - 2x_4)^3 + 2x_2 \cdot x_3 (4x_4^2 + 3x_1 \cdot (x_1 - 2x_4))} \leq 0$$

Where:

$$k_1 = 0.09755988; \quad k_2 = 0.99 \cdot k_1; \quad k_3 = 0.0391908; \quad k_4 = 0.9 \cdot k_3;$$

$$punishment = 10^3 \cdot \sum_{i=1}^2 \max(0, g_i(x))^2$$

Variable range:

$$10 \leq x_1 \leq 80; \quad 10 \leq x_2 \leq 50; \quad 0.9 \leq x_3 \leq 5; \quad 0.9 \leq x_4 \leq 5;$$

In this study, we conducted comparative tests between ESTGWOA and ZOA, ROA, GWO, AROA, HHO, MWOA, MSWOA, IPSO and WOA. The configuration parameters for all algorithms are detailed in Table 7. The iteration count was uniformly set at $T=500$, with a population size of $N=30$. To ensure reliability, each algorithm was executed independently 30 times on the I-beam design problem. The performance metrics, including *Ave* and *Std*, were recorded for evaluation. The experimental results are shown in Fig. 22 and Table 12. As shown in Table 12, ESTGWOA demonstrated significantly superior optimization accuracy and stability compared to other algorithms. This indicated that ESTGWOA had a substantial advantage in handling such problems.

Piston lever

The piston lever (PL) optimization problem, as shown in Fig. 23, focuses on designing a mechanical lever system that transmits force efficiently⁶⁵. The problem includes four design variables: x_1 and x_2 represent the primary length and width dimensions that define the overall geometry of the lever; x_3 denotes the radius of the cross-section at the force application point, which directly affects the stress distribution; and x_4 relates to the geometry around the support point. The objective is typically to minimize weight or maximize load-bearing efficiency while satisfying geometric and mechanical constraints.

The objective function for the piston lever design problem can be described as:

Variable:

$$x = [x_1, x_2, x_3, x_4] \tag{56}$$

Minimize:

$$y = 0.25\pi x_3^2(L_2 - L_1) + \text{punishment} \tag{57}$$

Subject to:

$$g_1(x) = QL \cos \theta - RF \leq 0; \tag{58}$$

$$g_2(x) = Q(L - x_4) - M_{\max} \leq 0; \tag{59}$$

$$g_3(x) = 1.2(L_2 - L_1) - L_1 \leq 0; \tag{60}$$

$$g_4(x) = \frac{x_3}{2} - x_2 \leq 0; \tag{61}$$

Variable range:

$$0.05 \leq x_1 \leq 500; \quad 0.05 \leq x_2 \leq 500; \quad 0.05 \leq x_4 \leq 500; \quad 0.05 \leq x_3 \leq 120; \tag{62}$$

Where:

$$\text{punishment} = 10^3 \cdot \sum_{i=1}^4 \max(0, g_i(x))^2$$

$$Q = 10000; \quad P = 1500; \quad L = 240; \quad M_{\max} = 1.8 \times 10^6; \tag{63}$$

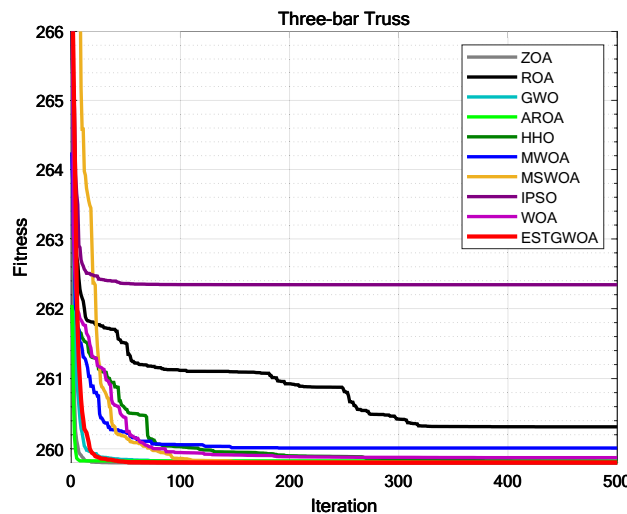


Fig. 26. Iteration curves of the algorithms in Three-bar Truss design.

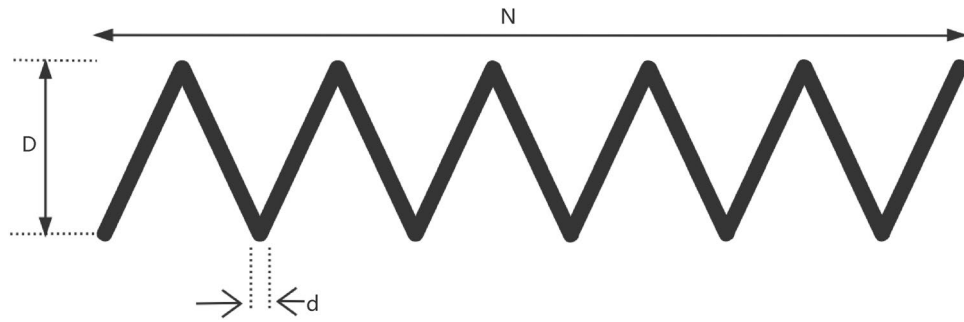


Fig. 27. The structure of a tension/compression spring.

$$L_1 = \sqrt{(x_4 - x_2)^2 + x_1^2}; \quad L_2 = \sqrt{(x_4 \sin \theta + x_1)^2 + (x_2 - x_4 \cos \theta)^2}; \quad (64)$$

$$R = \frac{|-x_4(x_4 \sin \theta + x_1) + x_1(x_2 - x_4 \cos \theta)|}{\sqrt{(x_4 - x_2)^2 + x_1^2}}; \quad (65)$$

$$F = 0.25\pi P x_3^2; \quad (66)$$

In this study, we conducted comparative tests between ESTGWOA and ZOA, ROA, GWO, AROA, HHO, MWOA, MSWOA, IPSO and WOA. The configuration parameters for all algorithms are detailed in Table 7. The iteration count was uniformly set at $T=500$, with a population size of $N=30$. To ensure reliability, each algorithm was executed independently 30 times on the piston lever design problem. The performance metrics, including *Ave* and *Std*, were recorded for evaluation. The experimental results were shown in Fig. 24, and Table 12. As shown in Table 12, in the piston lever design problem, ESTGWOA demonstrated significantly superior optimization accuracy and stability compared to other algorithms. This indicated that ESTGWOA had a substantial advantage in handling such problems.

Three-bar truss

The three-bar truss (TBT) design problem, as shown in Fig. 25, is a well-known benchmark in structural optimization, involving two continuous decision variables: the cross-sectional areas x_1 and x_2 ³⁶. The objective is to minimize the total volume of the truss under loading conditions. The problem includes a nonlinear objective function and three nonlinear inequality constraints that ensure stress and displacement limits are not violated. This compact yet challenging design problem serves as a standard test for evaluating the capabilities of MAs in handling nonlinearity and constraints.

The objective function for the three-bar truss design problem can be described as follows:

Variable:

$$x = [x_1, x_2]$$

Minimize:

$$y = (2\sqrt{2}x_1 + x_2) \cdot l + \text{punishment}$$

Subject to:

$$g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1 \cdot x_2} \cdot P - \sigma \leq 0;$$

$$g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1 \cdot x_2} \cdot P - \sigma \leq 0;$$

$$g_3(x) = \frac{1}{x_1 + \sqrt{2}x_2} \cdot P - \sigma \leq 0$$

Where:

$$\text{punishment} = 10^3 \cdot \sum_{i=1}^4 \max(0, g_i(x))^2$$

$$l = 100; \quad P = 2; \quad \sigma = 2;$$

Variable range:

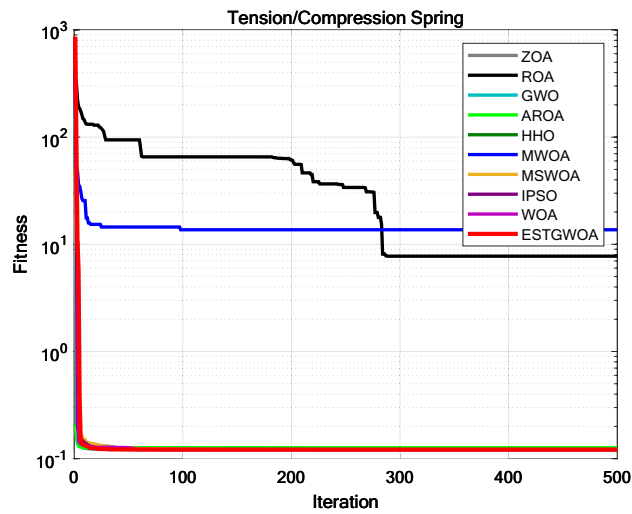


Fig. 28. Iteration curves of the algorithms in tension/compression spring design.

Full name	Abbreviation
Metaheuristic Algorithm	MA
State-of-the-art	SOTA
Good Nodes Set	GNS
Elite Guided Searching	EGS
Spiral-based Encircling Prey	SEP
Triangular-based Spiral Hunting	TSH
Differential Evolution	DE
Average fitness	Ave
Standard deviation	Std
Overall Effectiveness	OE
Multi-disk Clutch Brake	MDCB
Gas Transmission Compressor	GTC
Reactor Network	RNW
I-beam	IB
Piston lever	PL
Three-bar Truss	TBT
Tension/Compression Spring	TCS

Table 13. List of abbreviations and their corresponding full names used in this paper.

$$10 \leq x_1 \leq 80; \quad 10 \leq x_2 \leq 50; \quad 0.9 \leq x_3 \leq 5; \quad 0.9 \leq x_4 \leq 5;$$

In this study, we conducted comparative tests between ESTGWOA and ZOA, ROA, GWO, AROA, HHO, MWOA, MSWOA, IPSO and WOA. The configuration parameters for all algorithms are detailed in Table 7. The iteration count was uniformly set at $T=500$, with a population size of $N=30$. To ensure reliability, each algorithm was executed independently 30 times on the three-bar truss design problem. The performance metrics, including *Ave* and *Std*, were recorded for evaluation. The experimental results were shown in Fig. 26 and Table 12. As shown in Table 12, ESTGWOA demonstrated significantly superior optimization accuracy and stability compared to other algorithms. This indicated that ESTGWOA had a substantial advantage in handling such problems.

Tension/compression spring

The tension/compression spring (TCS) design problem, as shown in Fig. 27, is a classical constrained engineering task widely used in mechanical design^{35,36,66}. It involves three decision variables: the wire diameter (d), the mean coil diameter (D), and the number of active coils (N).

- d : the wire diameter;
- D : the mean coil diameter;
- N : the number of active coils.

The objective is to minimize the spring's weight while maintaining structural integrity and mechanical performance. Four nonlinear constraints, denoted as $g_1(x)$ to $g_4(x)$, are used to enforce limits on shear stress, free length, deflection, and surge frequency. These constraints ensure that the spring remains safe and functional under operational conditions. The objective function for the tension/compression spring design problem can be described as:

Variable:

$$x = [d, D, N] = [x_1, x_2, x_3]$$

Minimize:

$$y = (x_3 + 2) \cdot x_2 \cdot x_1^2 + \text{punishment} \quad (67)$$

Subject to:

$$g_1(x) = 1 - \frac{x_2^3 \cdot x_3}{71785x_1^4} \leq 0 \quad (68)$$

$$g_2(x) = \frac{4x_2^2 - x_1 \cdot x_2}{12566(x_2 \cdot x_1^3 - x_4)} + \frac{1}{5108x_1^2} - 1 \leq 0 \quad (69)$$

$$g_3(x) = 1 - \frac{140.45x_1}{x_2^2 \cdot x_3} \leq 0 \quad (70)$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \quad (71)$$

Where:

$$\text{punishment} = 10^3 \cdot \sum_{i=1}^4 \max(0, g_i(x))^2$$

Variable range:

$$0.05 \leq x_1 \leq 2, \quad 0.25 \leq x_2 \leq 1.3, \quad 2.0 \leq x_3 \leq 15$$

In this study, we conducted comparative tests between ESTGWOA and ZOA, ROA, GWO, AROA, HHO, MWOA, MSWOA, IPSO and WOA. The configuration parameters for all algorithms are detailed in Table 7. The iteration count was uniformly set at $T=500$, with a population size of $N=30$. To ensure reliability, each algorithm was executed independently 30 times on the tension/compression spring design problem. The performance metrics, including *Ave* and *Std*, were recorded for evaluation. The experimental results are presented in Fig. 28, and Table 12. As shown in Table 12, ESTGWOA demonstrated significantly superior optimization accuracy and stability compared to other algorithms. This indicated that ESTGWOA had a substantial advantage in handling such problems.

It is worth noting that although classical optimization techniques (e.g., nonlinear programming, penalty methods) have been applied to these problems in prior studies, their performance is often limited due to the highly non-convex and constrained search spaces. In contrast, ESTGWOA demonstrates robust optimization accuracy and stability across all cases, confirming its suitability for complex engineering applications.

Discussion

ESTGWOA effectively addresses several limitations of the canonical WOA, including premature convergence, low population diversity in later iterations, slow convergence speed, low convergence accuracy, and an imbalance between exploration and exploitation. As demonstrated by the experimental results, ESTGWOA exhibits strong competitiveness when compared with other state-of-the-art optimization algorithms.

From a theoretical perspective, the time complexity analysis presented in Section 4.7 shows that ESTGWOA and WOA share the same asymptotic time complexity, i.e., $O(T * ND)$. However, in practice, ESTGWOA generally requires slightly more computational time than the canonical WOA due to the introduction of the Hybrid Gaussian Mutation based on Differential Evolution strategy, which increases the number of arithmetic operations and fitness evaluations per iteration. This additional computational cost represents a trade-off for improved convergence accuracy, robustness, and population diversity.

Consequently, while ESTGWOA may not be the most suitable choice for large-scale real-time optimization tasks with strict time constraints, it demonstrates strong potential for solving complex real-world optimization problems. Its application scope includes mechanical design optimization, path planning, production scheduling, and neural network parameter tuning, where high-dimensional search spaces and practical constraints are commonly encountered. In addition, with appropriate encoding schemes and solution representation, ESTGWOA can be extended to address discrete and combinatorial optimization problems, such as feature selection, job-shop scheduling, and routing problems. ESTGWOA is particularly well suited for problems requiring a balanced exploration-exploitation mechanism, such as three-dimensional traveling salesman problems, manufacturing process simulation, and multi-objective engineering design. Future work will focus on deploying ESTGWOA

in practical industrial environments, exploring discrete and combinatorial variants, and further improving its computational efficiency under dynamic and uncertain conditions.

Abbreviations and their corresponding full names used in this paper are listed in Table 13.

Conclusion

ESTGWOA introduced Good Nodes Set Initialization to generate a uniformly distributed whale population. It employed newly-designed strategies including the Elite-guided Searching strategy, Spiral-based Encircling Prey strategy, Triangular-based Spiral Hunting strategy, and Hybrid Gaussian Mutation based on Differential Evolution. ESTGWOA adopted the newly-designed convergence factor a to balance global exploration and local exploitation. Experimental results demonstrated that ESTGWOA effectively balances exploration and exploitation during the optimization process, achieving high convergence accuracy, fast convergence, and maintaining population diversity.

To validate the effectiveness of the improvements in ESTGWOA, distinct tests were conducted, including Parameter sensitivity analysis experiment, Ablation study, Comparative experiment with SOTA algorithms, to verify ESTGWOA's superiority. And ESTGWOA was applied to seven engineering design optimization challenges, verifying its feasibility in engineering design optimization. ESTGWOA provided a novel approach for the application of WOA in engineering design.

Data availability

To support the experimental study in this paper, we used the Standard Benchmark Functions. The relevant data has been uploaded to Figshare, and the link for the specific modeling of Standard Benchmark Functions is: <https://doi.org/10.6084/m9.figshare.28440863>, only for reference and further analysis by the readers. The specific modeling of engineering optimization challenges in this study has been uploaded to Figshare, and the link is: https://figshare.com/articles/thesis/engineering_m/28673777?file=53256305, only for reference and further analysis by the readers.

Code availability

To support the experimental study in this paper, we used the Standard Benchmark Functions. The relevant data has been uploaded to Figshare, and the link for the specific modeling of Standard Benchmark Functions is: <https://doi.org/10.6084/m9.figshare.28440863>, only for reference and further analysis by the readers⁶⁷. The specific modeling of engineering optimization challenges in this study has been uploaded to Figshare, and the link is: <https://doi.org/10.6084/m9.figshare.28673777.v1>, only for reference and further analysis by the readers.

Received: 14 November 2025; Accepted: 21 January 2026

Published online: 12 February 2026

References

- Holland, J. H. Genetic algorithms. *Sci. Am.* **267**, 66–73 (1992).
- Das, S. & Suganthan, P. N. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **15**, 4–31 (2010).
- Xue, J. & Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* **8**, 22–34 (2020).
- Gao, Y., Wang, J. & Li, C. Escape after love: Philoponella prominens optimizer and its application to 3d path planning. *Clust. Comput.* **28**, 81 (2025).
- Gámez, M. G. M. & Vázquez, H. P. A novel swarm optimization algorithm based on hive construction by tetragonula carbonaria builder bees. *Mathematics* **13**, 2721 (2025).
- Agushaka, J. O. et al. Greater cane rat algorithm (gcra): A nature-inspired metaheuristic for optimization problems. *Heliyon* **10**, e31629 (2024).
- Dorigo, M., Birattari, M. & Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **1**, 28–39 (2007).
- Kennedy, J. & Eberhart, R. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks* vol. 4, 1942–1948 (IEEE, 1995).
- Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014).
- Heidari, A. A. et al. Harris hawks optimization: Algorithm and applications. *Futur. Gener. Comput. Syst.* **97**, 849–872 (2019).
- Jia, H., Peng, X. & Lang, C. Remora optimization algorithm. *Expert Syst. Appl.* **185**, 115665 (2021).
- Trojovská, E., Dehghani, M. & Trojovský, P. Zebra optimization algorithm: A new bio-inspired optimization algorithm for solving optimization algorithm. *Ieee Access* **10**, 49445–49473 (2022).
- Fu, S. et al. Red-billed blue magpie optimizer: A novel metaheuristic algorithm for 2d/3d uav path planning and engineering design problems. *Artif. Intell. Rev.* **57**, 134 (2024).
- Hamadneh, T. et al. Salamander optimization algorithm: A new bio-inspired approach for solving optimization problems. *Int. J. Intell. Eng. Syst* **18**, 550–561 (2025).
- Wang, X. Bighorn sheep optimization algorithm: A novel and efficient approach for wireless sensor network coverage optimization. *Phys. Scr.* <https://doi.org/10.1088/1402-4896/ade378> (2025).
- Alibabaei Shahraki, M. Cloud drift optimization algorithm as a nature-inspired metaheuristic. *Discov. Comput.* **28**, 173 (2025).
- Hatamlou, A. Black hole: A new heuristic optimization approach for data clustering. *Inf. Sci.* **222**, 175–184 (2013).
- Mirjalili, S., Mirjalili, S. M. & Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **27**, 495–513 (2016).
- Hashim, F. A., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W. & Mirjalili, S. Henry gas solubility optimization: A novel physics-based algorithm. *Futur. Gener. Comput. Syst.* **101**, 646–667 (2019).
- Van Laarhoven, P. J. & Aarts, E. H. Simulated annealing. In *Simulated Annealing: Theory and Applications* 7–15 (Springer, 1987).

21. Cymerys, K. & Oszust, M. Attraction-repulsion optimization algorithm for global optimization problems. *Swarm Evol. Comput.* **84**, 101459 (2024).
22. Zraiqat, A. et al. Thunderstorm and cloud algorithm: A novel parameter-free metaheuristic inspired by atmospheric dynamics for complex optimization tasks. *Int. J. Intell. Eng. Syst.* **18**, 153–167 (2025).
23. Hussein, N. K. et al. Schrödinger optimizer: A quantum duality-driven metaheuristic for stochastic optimization and engineering challenges. *Knowl.-Based Syst.* **328**, 114273 (2025).
24. Zraiqat, A. et al. Library and readers algorithm (lra): A novel human-inspired parameter-free metaheuristic for efficient global optimization. *Int. J. Intell. Eng. Syst.* **18**, 526–540 (2025).
25. Zraiqat, A. et al. Driver and navigator algorithm: A novel parameter-free human-inspired metaheuristic for efficient global optimization. *Int. J. Intell. Eng. Syst.* **18**, 555–569 (2025).
26. Zraiqat, A. et al. Court and judge algorithm (cja): A novel human-inspired metaheuristic for engineering optimization. *Int. J. Intell. Eng. Syst.* **18**, 60–72 (2025).
27. Zraiqat, A. et al. Community-based crisis management algorithm (ccma): A novel parameter-free metaheuristic for complex constrained optimization. *Int. J. Intell. Eng. Syst.* **18**, 593–606 (2025).
28. Rao, R. V., Savsani, V. J. & Vakharia, D. P. Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Comput. Aided Des.* **43**, 303–315 (2011).
29. Moosavian, N. & Roodsari, B. K. Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks. *Swarm Evol. Comput.* **17**, 14–24 (2014).
30. Zraiqat, A. et al. Psychologist algorithm: A human-inspired metaheuristic for solving complex constrained optimization problems. *Int. J. Intell. Eng. Syst.* **18**, 124–137 (2025).
31. Hamadneh, T. et al. Perfumer optimization algorithm: A novel human-inspired metaheuristic for solving optimization tasks. *Int. J. Intell. Eng. Syst.* **18**, 633–643 (2025).
32. Wei, J. et al. An enhanced whale optimization algorithm with log-normal distribution for optimizing coverage of wireless sensor networks. arXiv preprint [arXiv:2511.15970](https://arxiv.org/abs/2511.15970) (2025).
33. Wang, X. & Yao, L. Cape lynx optimizer: A novel metaheuristic algorithm for enhancing wireless sensor network coverage. *Measurement* **256**, 118361 (2025).
34. Paredes, M., Sartor, M. & Masclot, C. An optimization process for extension spring design. *Comput. Methods Appl. Mech. Eng.* **191**, 783–797 (2001).
35. Wei, J. et al. Lsewoa: An enhanced whale optimization algorithm with multi-strategy for numerical and engineering design optimization problems. *Sensors* **25**, 2054 (2025).
36. Wei, J., Gu, Y., Lu, B. & Cheong, N. Rwoa: A novel enhanced whale optimization algorithm with multi-strategy for numerical optimization and engineering design problems. *PLoS ONE* **20**, e0320913 (2025).
37. Dey, V. et al. Optimization of bead geometry in electron beam welding using a genetic algorithm. *J. Mater. Process. Technol.* **209**, 1151–1157 (2009).
38. Gu, Y., Wei, J. & Cheong, N. Credit card fraud detection based on minikm-svmsmote-xgboost model. In *Proceedings of the 2024 8th International Conference on Big Data and Internet of Things* 252–258 (2024).
39. Agrawal, U. K. & Panda, N. Quantum-inspired adaptive mutation operator enabled pso (qamo-pso) for parallel optimization and tailoring parameters of kolmogorov-arnold network. *J. Supercomput.* **81**, 1310 (2025).
40. Agrawal, U. K., Panda, N., Tejani, G. G. & Mousavirad, S. J. Improved salp swarm algorithm-driven deep cnn for brain tumor analysis. *Sci. Rep.* **15**, 24645 (2025).
41. Wei, J. et al. Nawoa-xgboost: A novel model for early prediction of academic potential in computer science students. arXiv preprint [arXiv:2512.04751](https://arxiv.org/abs/2512.04751) (2025).
42. Qaraad, M. et al. Comparing ssaleo as a scalable large scale global optimization algorithm to high-performance algorithms for real-world constrained optimization benchmark. *IEEE Access* **10**, 95658–95700 (2022).
43. Mahapatra, A. K., Panda, N. & Pattanayak, B. K. Quantized orthogonal experimentation ssa (qox-ssa): A hybrid technique for feature selection (fs) and neural network training. *Arab. J. Sci. Eng.* **50**, 1025–1056 (2025).
44. Mahapatra, A. K., Panda, N. & Pattanayak, B. K. Adaptive dimensional search-based orthogonal experimentation ssa (adox-ssa) for training rbf neural network and optimal feature selection. *J. Supercomput.* **81**, 212 (2025).
45. Mahapatra, A. K., Panda, N., Mahapatra, M., Jena, T. & Mohanty, A. K. A fast-flying particle swarm optimization for resolving constrained optimization and feature selection problems. *Clust. Comput.* **28**, 91 (2025).
46. Levi, Y., Bekhor, S. & Rosenfeld, Y. A multi-objective optimization model for urban planning: The case of a very large floating structure. *Transp. Res. Part C* **98**, 85–100 (2019).
47. Wei, J. et al. Ahrtr: An enhanced rapidly-exploring random tree algorithm with heuristic search for uav urban path planning. *Preprints:2025111805* (2025).
48. Wei, J., Gu, Y., Law, K. E. & Cheong, N. Adaptive position updating particle swarm optimization for uav path planning. In *2024 22nd International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)* 124–131 (IEEE, 2024).
49. Xie, Z. et al. Research on uav applications in public administration: Based on an improved rrt algorithm. arXiv preprint [arXiv:2508.14096](https://arxiv.org/abs/2508.14096) (2025).
50. Lu, B. et al. Mrbmo: An enhanced red-billed blue magpie optimization algorithm for solving numerical optimization challenges. *Symmetry* **17**, 1295 (2025).
51. Käsichel, J., Teich, T. & Zacher, B. Real-time dynamic shop floor scheduling using evolutionary algorithms. *Int. J. Prod. Econ.* **79**, 113–120 (2002).
52. Mahmood, B. S., Hussein, N. K., Aljohani, M. & Qaraad, M. A modified gradient search rule based on the quasi-newton method and a new local search technique to improve the gradient-based algorithm: solar photovoltaic parameter extraction. *Mathematics* **11**, 4200 (2023).
53. Qaraad, M. et al. Photovoltaic parameter estimation using improved moth flame algorithms with local escape operators. *Comput. Electr. Eng.* **106**, 108603 (2023).
54. Qaraad, M. et al. Quadratic interpolation and a new local search approach to improve particle swarm optimization: Solar photovoltaic parameter estimation. *Expert Syst. Appl.* **236**, 121417 (2024).
55. Gao, Y. & Xie, S.-L. Chaos particle swarm optimization algorithm. *Comput. Sci.* **31**, 13–15 (2004).
56. Qu, C., He, W., Peng, X. & Peng, X. Harris hawks optimization with information exchange. *Appl. Math. Model.* **84**, 52–75 (2020).
57. Jia, H. et al. Improved snow ablation optimizer with heat transfer and condensation strategy for global optimization problem. *J. Comput. Des. Eng.* **10**, 2177–2199 (2023).
58. Huang, J. & Hu, H. Hybrid beluga whale optimization algorithm with multi-strategy for functions and engineering optimization problems. *J. Big Data* **11**, 3 (2024).
59. Mirjalili, S. & Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67 (2016).
60. Liu, M., Yao, X. & Li, Y. Hybrid whale optimization algorithm enhanced with lévy flight and differential evolution for job shop scheduling problems. *Appl. Soft Comput.* **87**, 105954 (2020).
61. Chakraborty, S., Sharma, S., Saha, A. K. & Saha, A. A novel improved whale optimization algorithm to solve numerical optimization and real-world applications. *Artif. Intell. Rev.* **55**, 4605–4716 (2022).
62. Li, C. et al. Evolving the whale optimization algorithm: The development and analysis of miswoa. *Biomimetics* **9**, 639 (2024).

63. Gu, Y. et al. Gwoa: A multi-strategy enhanced whale optimization algorithm for engineering design optimization. *PLoS ONE* **20**, e0322494 (2025).
64. Xiao, C., Cai, Z. & Wang, Y. A good nodes set evolution strategy for constrained optimization. In *2007 IEEE Congress on Evolutionary Computation* 943–950 (IEEE, 2007).
65. Wei, J. et al. Tswoa: An enhanced woa with triangular walk and spiral flight for engineering design optimization. In *2025 8th International Conference on Advanced Algorithms and Control Engineering (ICAACE)* 186–194 (IEEE, 2025).
66. Wei, J. et al. Lswoa: An enhanced whale optimization algorithm with levy flight and spiral flight for numerical and engineering design optimization problems. *PLoS ONE* **20**, e0322058 (2025).
67. Suganthan, P. N. et al. Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. *KanGAL Rep.* **2005005**, 2005 (2005).
68. Anitha, J., Pandian, S. I. A. & Agnes, S. A. An efficient multilevel color image thresholding based on modified whale optimization algorithm. *Expert Syst. Appl.* **178**, 115003 (2021).
69. Yang, W. et al. A multi-strategy whale optimization algorithm and its application. *Eng. Appl. Artif. Intell.* **108**, 104558 (2022).
70. Nadimi-Shahraki, M. H., Taghian, S., Mirjalili, S. & Faris, H. Mtd: An effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems. *Appl. Soft Comput.* **97**, 106761 (2020).

Acknowledgements

The supports provided by Macao Polytechnic University (MPU Grant no: RP/FCA-03/2022; RP/FCA-04/2022; RP/FCA-06/2022; RP/FCA-01/2025) and MacaoScience and Technology Development Fund (FDCT Grant no: 0044/2023/ITP2; FDCT-MOST (0018/2025/AMJ)) enabled us to conduct data collection, analysis, and interpretation, as well as cover expenses related to research materials and participant recruitment. MPU and FDCT investment in our work (MPU Submission Code: fca.def5.c31c.0) have significantly contributed to the quality and impact of our research findings.

Author contributions

J.W. conceived and designed the study; J.W. developed the methodology and implemented the software; J.W. performed the validation; R.Z., S.W., Z.L. and W.Z. conducted the formal analysis; J.W. carried out the investigation; Z.W., N.C., S.-K.I., Y.W. and X.Y. provided the resources; W.Z., S.W., Z.L., J.W., Y.L., Y.G., and R.Z. performed the data curation; J.W. prepared the original draft; Z.W., N.C., S.-K.I., Y.W. and X.Y. reviewed and edited the manuscript; W.Z., S.W., Z.L., J.W., Y.L., Y.G., and R.Z. contributed to the visualization; Z.W., N.C., S.-K.I., Y.W. and X.Y. supervised the research and managed the project; N.C., Y.W. and X.Y. acquired the funding. All authors reviewed the manuscript.

Funding

This work is supported by the grant from FDCT-MOST (0018/2025/AMJ) and Macao Polytechnic University (RP/FCA-01/2025).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to X.Y.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2026