Journal of Zhejiang University-SCIENCE A 2025 26(12):1197-1210
www.jzus.zju.edu.cn; www.springer.com/journal/11582
E-mail: jzus_a@zju.edu.cn

Research Article
https://doi.org/10.1631/jzus.A2400512

Check for updates

Improved coati optimization algorithm through multi-strategy integration: from theoretical design to engineering applications

Shuangxi LIU[1], Ruizhe FENG[1], Yuxin WEI[2], Wei HUANG[1]✉, Binbin YAN[2]

[1]*Advanced Propulsion Technology Laboratory, National University of Defense Technology, Changsha 410073, China*
[2]*School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China*

Abstract: Optimization problems are crucial for a wide range of engineering applications, as efficient solutions lead to better performance. This study introduces an improved coati optimization algorithm (ICOA) that overcomes the primary limitations of the original coati optimization algorithm (COA), notably its insufficient population diversity and propensity to become trapped in local optima. To address these issues, the ICOA integrates three innovative strategies: Latin hypercube sampling (LHS), Lévy-flight, and an adaptive local search. LHS is employed to ensure a diverse initial population, thereby laying a foundation for the optimization. Lévy-flight is utilized to facilitate an efficient global search, enhancing the algorithm's ability to explore the solution space. The adaptive local search is designed to refine solutions, enabling more precise local exploration. Together, these strategies significantly improve the population's quality and diversity, thereby improving the algorithm's convergence accuracy and optimization capabilities. The performance of the ICOA is tested against several established algorithms, using 12 benchmark functions. Additionally, the ICOA's practicality and effectiveness are demonstrated through application to a real-world engineering problem, specifically the design optimization of tension/compression springs. Simulation results show that the ICOA consistently outperforms the other algorithms, providing robust solutions for a wide range of optimization problems.

Key words: Improved coati optimization algorithm (ICOA); Latin hypercube sampling (LHS); Lévy-flight; Adaptive local search; Multi-strategy; Engineering applications

1 Introduction

Optimization problems hold significant importance in engineering and other applied areas of study (Li et al., 2023; Ye et al., 2024). For example, in military operation planning, optimization methods are used to determine the ideal distribution of forces. This involves considering factors such as terrain, enemy capabilities, and strategic objectives to ensure maximum combat efficiency while minimizing losses (Yang et al., 2022; Shamami et al., 2024; Liu et al., 2025). Similarly, in weapons development, optimization is employed to enhance the performance of armaments. This might include improving the range, accuracy, and lethality of missiles, or the firepower and mobility of armored vehicles (Wade, 2019; Karimi et al., 2024; Zhao et al., 2024). Furthermore, optimization is vital for surveillance systems, where it is used to enhance the coverage, sensitivity, and data-processing speed of sensors, thereby enabling early threat detection (Cui et al., 2022; Sun et al., 2024; Wang CG et al., 2024).

Research on rapid solutions to these optimization problems has grown in importance (Pestana Barros, 2004). Rapid optimization solutions are crucial for providing immediate and effective responses to dynamic battlefield situations, thereby granting the military a decisive advantage (Heise and Morse, 2000). These solutions also facilitate efficient resource utilization within the defense sector, promoting optimal allocation of limited funds across various defense projects (Terziev and Nichev, 2017). However, as the size and complexity of related problems increase, traditional deterministic optimization algorithms often fail to meet practical requirements (Martí Sempere, 2023). Consequently, heuristic algorithms have gained prominence

✉ Wei HUANG, gladrain2001@163.com
Shuangxi LIU, https://orcid.org/0000-0002-1422-1096
Ruizhe FENG, https://orcid.org/0009-0003-1941-3024
Wei HUANG, https://orcid.org/0000-0001-9805-985X

Received Oct. 31, 2024; Revision accepted Jan. 19, 2025;
Crosschecked Nov. 21, 2025

and have been extensively developed over the past few decades to address these challenges.

Swarm intelligence algorithms have garnered significant attention among heuristic algorithms due to their capacity to emulate the collective behavior observed in natural organisms (Brezočnik et al., 2018; Hashim et al., 2021; Saeed et al., 2022; Sharma et al., 2022; Agushaka et al., 2023). By simulating processes in a manner similar to interactions and behaviors of animals, insects, or other organisms, these algorithms effectively navigate the search space to identify optimal solutions. Notable examples of swarm intelligence algorithms include particle swarm optimization (PSO) (Kennedy and Eberhart, 1995; Wang et al., 2018), ant lion optimization (ALO) (Mirjalili, 2015a; Ali et al., 2017), grey wolf optimization (GWO) (Mirjalili et al., 2014; Faris et al., 2018), and moth flame optimization (MFO) (Mirjalili, 2015b; Sahoo and Saha, 2022). These algorithms have consistently demonstrated strong performance and adaptability in addressing a wide range of complex optimization problems.

Despite the significant achievements of swarm intelligence algorithms, ongoing research has revealed several limitations inherent to these methods. These limitations include a tendency to fall into local optima, slow convergence speeds, and difficulty in parameter adjustment (Bingul and Karahan, 2018; Liu et al., 2022). Consequently, enhancing the performance and effectiveness of swarm intelligence algorithms, either through development of new algorithms or refinements to existing ones, has become a prominent research area in recent years (Song et al., 2019; Zhao et al., 2025).

By Liu et al. (2021, 2022) and Ma et al. (2023), a simulated annealing algorithm was combined with adaptive PSO to effectively address the issue of the PSO becoming trapped in local optima. This approach has been successfully applied in engineering fields such as multi-target allocation, multi-missile formation design, and multi-missile cooperative detection. To address route planning challenges for unmanned aerial vehicles, Yao and Wang (2017) developed a dynamic adaptive ant lion optimizer (DAALO). This approach replaces the traditional random walk of ants with Lévy flight, thereby enhancing the ALO's ability to escape local optima. Furthermore, DAALO incorporates a feedback mechanism based on the improvement rate of the population, which dynamically adjusts the size of the trap according to the 1/5 principle. This adjustment significantly enhances the performance of the ALO, improving its convergence accuracy, convergence speed, and stability. In the context of agricultural drone trajectory planning, Liu et al. (2023) introduced a multi-strategy collaborative improvement GWO algorithm. This algorithm enhances search accuracy by employing an evolutionary boundary constraint processing mechanism. This mechanism updates the positions of grey wolf individuals that exceed boundaries, thereby preserving the positional information of the optimal individual. Cui et al. (2020) developed an enhanced MFO algorithm incorporating an adaptive Lévy flight strategy to optimize the external parameters of variable cycle engines. By evaluating the fitness variance difference between consecutive generations of moths, this algorithm effectively assesses the population aggregation state, thereby improving the global search capability.

The coati optimization algorithm (COA) is a novel swarm intelligence optimization technique inspired by the foraging behavior of South American coatis, known for their complex group cooperation and highly social nature (Dehghani et al., 2023). This algorithm emulates the coatis' collaborative search for food by facilitating information sharing and cooperation among individuals, enabling identification of optimal solutions. The core principle of COA is to balance "exploration" and "utilization." Exploration refers to extensively searching the solution space to uncover potential solutions, while "utilization" refers to making precise adjustments near known high-quality solutions to enhance their accuracy. This dual approach enables COA to tackle complex optimization challenges, including function optimization, feature selection, and scheduling.

In (Wang Y et al., 2024), COA was selected to optimize the control parameters of the load frequency controller in power systems, utilizing four error integration criteria. Their simulation results showed how the approach effectively achieves load frequency control by significantly reducing the maximum frequency deviation and maintaining frequency stability. In (Le et al., 2024), COA was employed to create adaptive synthesis rules for low-frequency components in medical image synthesis. This approach improves image quality, brightness, and contrast while preserving essential details like boundaries, edges, and the original image structure. Li et al. (2024) approached the problem

of low detection accuracy in low-cost gas sensors affected by external disturbances by introducing a modified raccoon gated recurrent unit neural network model (COA-GRU), which notably enhances the sensors' detection accuracy.

Although COA has demonstrated advantages over other typical swarm intelligence algorithms in solving optimization problems, it also faces two significant limitations that restrict its effectiveness in practical scenarios (Hashim et al., 2023; Emam et al., 2024).

(1) COA suffers from a lack of population diversity. The random generation of the initial population may result in insufficient diversity, which can impede the algorithm's capacity to effectively explore the solution space. This limitation increases the likelihood of premature convergence to suboptimal solutions, thereby hindering the search for the optimal solution.

(2) COA is prone to becoming stuck in local optima. Like many heuristic algorithms, COA may experience a slowdown in convergence speed and is susceptible to becoming trapped in local optima, particularly when applied to high-dimensional problems. These factors collectively constrain its breadth and effectiveness in practical applications.

To address the aforementioned limitations of COA, we integrate three strategies: Latin hypercube sampling (LHS), Lévy-flight, and adaptive local search, and accordingly propose an improved coati optimization algorithm (ICOA). This multi-strategy approach enhances the quality and diversity of the population, thereby improving the algorithm's convergence accuracy and optimization capabilities.

2 Overview and analysis of COA

2.1 Overview of COA

The COA, inspired by the hunting behavior of coatis, is a swarm intelligence method developed to solve optimization problems. It simulates two primary behaviors observed in coatis: hunting and attacking iguanas, and escaping from predators. By integrating these behaviors, the algorithm addresses complex optimization challenges through three collaborative phases: population initialization, exploration, and exploitation. This section offers a detailed introduction to COA and analyzes the mechanisms involved in each stage.

2.1.1 Phase 1: initialization

The initialization phase primarily aims to randomly generate an initial population within the optimized space, serving as the starting point for the COA's search. This phase is crucial as it establishes the foundation for the algorithm's execution, thereby facilitating the subsequent exploration and development phases. The main mathematical models are described as follows:

$$\boldsymbol{X}=\begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \cdots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,m} \\ \vdots & \cdots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,m} \end{bmatrix}_{N\times m}, \quad (1)$$

$$\begin{aligned} x_{i,j} &= l_{\mathrm{b}j} + r\cdot(u_{\mathrm{b}j} - l_{\mathrm{b}j}), \\ i &= 1,\ 2,\ \cdots,\ N,\ j = 1,\ 2,\ \cdots,\ m, \end{aligned} \quad (2)$$

$$\boldsymbol{F}=\begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1}, \quad (3)$$

where $\boldsymbol{X}$ denotes a vector with the initial positions of the coati population; $N \in \mathbb{N}^+$ denotes the population size of coatis, $m \in \mathbb{N}^+$ is the number of dimensions in the problem space; $r \in [0, 1]$ is a random variable; i and j denote the number and dimension, respectively; $\boldsymbol{F}$ denotes the objective function vector of the coati population position; $l_{\mathrm{b}j}$ and $u_{\mathrm{b}j}$ denote the lower and upper bounds of the jth dimensional variable, respectively.

2.1.2 Phase 2: exploration

During this stage, a group of coatis climbs a tree to encircle and intimidate the iguanas, while other coatis remain on the ground, ready to act. When an iguana falls from the tree, the coatis on the ground promptly pursue and attack it. This coordinated strategy exemplifies the coatis' ability to navigate different locations in their environment. Such behavior underscores the algorithm's global search and exploration capabilities, which are crucial for problem solving.

Specially, the model for updating the coatis' positions when climbing trees can be expressed as:

$$X_i^{\mathrm{P1}}: x_{i,j}^{\mathrm{P1}} = x_{i,j} + r \cdot (I_{\mathrm{g}j} - I x_{i,j}),$$
$$\text{for } i = 1,\ 2,\ \cdots,\ \left\lfloor \frac{N}{2} \right\rfloor,\ j = 1,\ 2,\ \cdots,\ m, \tag{4}$$

where the superscript P1 denotes the physical quantity during the exploration phase; X_i^{P1} is the position of the coati during its movement; I_{g} is the iguana's position in the search space; I is an integer randomly selected from the set $\{1, 2\}$; $[\cdot]$ denotes the floor function.

When the iguana falls from the tree, the coatis' position model under the tree is updated by

$$X_i^{\mathrm{P1}}: x_{i,j}^{\mathrm{P1}} = \begin{cases} x_{i,j} + r \cdot (I_{\mathrm{g}j}^{\mathrm{G}} - I x_{i,j}), & F_{\mathrm{Ig}}^{\mathrm{G}} < F_i, \\ x_{i,j} + r \cdot (x_{i,j} - I_{\mathrm{g}j}^{\mathrm{G}}), & F_{\mathrm{Ig}}^{\mathrm{G}} \geqslant F_i, \end{cases}$$
$$\text{for } i = \left\lfloor \frac{N}{2} \right\rfloor + 1,\ \left\lfloor \frac{N}{2} \right\rfloor + 2,\ \cdots,\ N,\ j = 1,\ 2,\ \cdots,\ m, \tag{5}$$

where the superscript G denotes the physical quantity during ground exploration, and I_{g} is the random position of the iguana on the ground.

If the coati's updated individual objective function value improves, its current position information is updated; otherwise, it is left unchanged.

$$X_i = \begin{cases} X_i^{\mathrm{P1}}, & F_i^{\mathrm{P1}} < F_i, \\ X_i, & F_i^{\mathrm{P1}} \geqslant F_i. \end{cases} \tag{6}$$

2.1.3 Phase 3: exploitation

This stage highlights the interactions between the coatis and their predators, focusing on the encounter and the coatis' subsequent escape actions. When a predator attacks, the coatis update their position within the search space by fleeing in a safe direction. This behavior exemplifies the COA's ability to formulate a local search strategy. The relevant mathematical models are described as follows:

$$l_{\mathrm{b}j}^{\mathrm{local}} = \frac{l_{\mathrm{b}j}}{t},\ u_{\mathrm{b}j}^{\mathrm{local}} = \frac{u_{\mathrm{b}j}}{t},\ t = 1,\ 2,\ \cdots,\ T, \tag{7}$$

$$X_i^{\mathrm{P2}}: x_{i,j}^{\mathrm{P2}} = x_{i,j} + (1 - 2r) \cdot \left(l_{\mathrm{b}j}^{\mathrm{local}} + r \cdot (u_{\mathrm{b}j}^{\mathrm{local}} - l_{\mathrm{b}j}^{\mathrm{local}}) \right),$$
$$\text{for } i = 1,\ 2,\ \cdots,\ N,\ j = 1,\ 2,\ \cdots,\ m, \tag{8}$$

where the superscript P2 denotes the physical quantity during the exploitation phase; the superscript "local" denotes the physical quantity corresponding to the local search space; $t \in \mathbb{N}^+$ is the current iteration count; $T \in \mathbb{N}^+$ is the maximum iteration count; X_i^{P2} refers to the position of the ith coati during the exploitation phase.

If the updated coati has an improved objective function value, the individual location information is updated; otherwise, it remains unchanged.

$$X_i = \begin{cases} X_i^{\mathrm{P2}}, & F_i^{\mathrm{P2}} < F_i, \\ X_i, & F_i^{\mathrm{P2}} \geqslant F_i. \end{cases} \tag{9}$$

2.2 Limitations of COA

Despite its simple structure and ease of implementation, COA exhibits several limitations that impede optimal performance.

(1) As shown in Eqs. (1) and (2), the initial population of COA lacks effective communication and interaction among individuals. This absence of diversity reduces the performance of the optimization algorithm. The initially generated population is often random, and this randomness can result in insufficient diversity during the search for the optimal solution. Consequently, this reduces the algorithm's effectiveness in exploring the solution space and increases the risk of premature convergence to suboptimal solutions.

(2) COA frequently shows a tendency to converge prematurely to local optima during the search for optimal solutions. This premature convergence, often stemming from prolonged random searching during the exploration phase, introduces heightened randomness and uncertainty. Additionally, COA may face challenges such as an inadequate balance between exploration and exploitation, sluggish convergence rates, and suboptimal convergence accuracy. These limitations are particularly pronounced in high-dimensional, complex, and nonlinear optimization problems, underscoring the need for improvements to COA's efficiency and effectiveness.

Although COA demonstrates a commendable balance between global and local search, and is characterized by strong evolutionary capabilities, rapid search speed, and robust optimization ability, it still requires enhancements for practical applications. As the complexity of optimization problems escalates (particularly in high-dimensional, complex, or nonlinear contexts and cases), the performance of COA may be impeded. Consequently, refinements to COA are essential for enhancing its efficiency and effectiveness in complex optimization challenges, and to expand its scope of potential applications.

3 Design of ICOA

Our proposed algorithm focuses on improving two key aspects: enhancing population diversity, and overcoming the tendency to become trapped in local optima. To design ICOA, we employ three strategies: LHS, Lévy flight, and adaptive local search.

3.1 LHS

LHS is a highly efficient multidimensional stratified sampling method that is extensively employed in uncertainty analysis and optimization. The primary objective of LHS is to achieve a more uniform coverage of the entire parameter space, which in turn enhances the efficiency of analyzing and simulating complex systems (Rajabi et al., 2015; Shields and Zhang, 2016; Zhang et al., 2020; Politis et al., 2021). In comparison to simple random sampling, LHS provides superior coverage of the parameter space and requires fewer samples.

Specifically, in the context of the population initialization stage in a swarm optimization algorithm, using LHS can significantly enhance the uniformity and diversity of the initial population's distribution across the search space, thereby establishing a solid foundation for the ensuing optimization process. The main process can be described as follows:

(1) Define the problem dimensions and sample size. Consider a problem of dimension m, and determine the number of samples n, which denotes the size of the population.

(2) Divide each dimension into strata. For each dimension, divide the range from l_{bi} to u_{bi} into n equal-probability intervals or strata.

(3) Generate random permutations. For each dimension, generate a random permutation of the integers from 1 to n.

(4) Sample within each stratum. For each dimension and sample index, calculate the sample value x_{ij} within the jth stratum.

(5) Map the results to the actual range of the bounds. After obtaining the samples x_{ij} using the LHS method, the values are adjusted to ensure they remain within the original bounds l_{bi} and u_{bi}, for each dimension i.

Remark 1 The application of LHS in population initialization ensures a more uniform distribution across the search space. This uniformity facilitates comprehensive exploration, thereby reducing the risk of overlooking optimal regions. Additionally, LHS increases population diversity, which helps to prevent premature convergence to suboptimal solutions. Consequently, the performance and effectiveness of the swarm optimization algorithm are enhanced, leading to more reliable results in complex optimization problems.

Next, we consider a population size of N=50 for the optimization process. The optimization variable is selected as $\boldsymbol{X}=[x_1, x_2]$, where the problem dimension is m=2. The domains for the variables are defined as $x_1\in[0, 10]$ and $x_2\in[5, 15]$. Fig. 1 presents a comparison of the population initialization results with and without LHS.

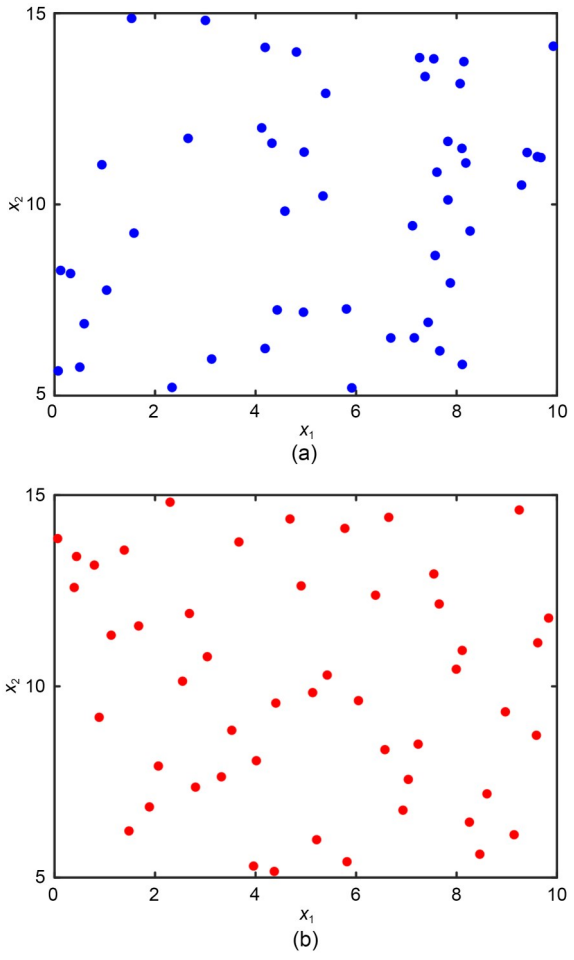

Fig. 1 Results of the population initialization: (a) without LHS; (b) with LHS

It can be observed that the randomly initialized population without LHS has a distribution that is more random and less uniform. Although this approach does

cover the entire space, the distribution of points is uneven, resulting in some areas being overly represented and others insufficiently sampled. In certain populations, this uneven distribution can lead to overlapping phenomena. Such irregularities can also cause search inefficiencies, as the algorithm may waste computational resources by exploring closely clustered regions. Consequently, it might also overlook promising areas due to the inherently random nature of the initialization.

Conversely, in the case of the population initialized via LHS, the point distribution is more uniformly scattered across the two-dimensional parameter space. This uniform dispersion is a crucial advantage of LHS. It implies that the sampling is more systematic and balanced across different regions of the search space. Consequently, it is more likely that a broader spectrum of solutions will be explored during the initial stages of the optimization process. This enhanced coverage can prevent the algorithm from prematurely converging to local optima, and augment the likelihood of finding the global optimum (or a closer approximation).

In summary, this comparison showcases the superiority of LHS in providing a more diverse and evenly distributed initial population. This characteristic is vital to the success of swarm optimization algorithms, as it establishes a better foundation for efficient exploration and convergence towards optimal solutions. It also highlights how employing appropriate sampling techniques like LHS in the initial phase can enhance the performance and effectiveness of optimization algorithms.

3.2 Lévy flight

In our proposed ICOA, the Lévy flight strategy is employed to enhance the algorithm's exploration capabilities, particularly during Phase 2. Lévy flight is a random walk process that significantly improves the ability to explore distant regions of the search space. Its application allows for a more thorough exploration of the broader solution space (Emary et al., 2019; Houssein et al., 2020; Iacca et al., 2021; He et al., 2023).

Specifically, when a certain condition is met in Phase 2 (a 50% probability of use), Lévy flight is utilized to update the coatis' positions. The mathematical model governing this update process is detailed as follows.

First, the Lévy step size is calculated. Set β=1.5 a parameter for the Lévy flight, and compute σ using:

$$\sigma=\left(\frac{\gamma(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\gamma\left(\frac{1+\beta}{2}\right)\beta\cdot 2^{\frac{1+\beta}{2}}}\right)^{\frac{1}{\beta}}, \tag{10}$$

where γ denotes the gamma function, defined as:

$$\gamma(z)=\int_0^{\infty}t^{z-1}\mathrm{e}^{-t}\mathrm{d}t. \tag{11}$$

Then, generate random Gaussian numbers u and v, matching the size of the position vector. The step size, which we refer to as step, is determined by $\varsigma=u/|v|^{\frac{1}{\beta}}$ Consequently, the updated position from Lévy flight is expressed as:

$$x_{i,j}^{\mathrm{P1,new}}=x_{i,j}+\varsigma\cdot(I_{\mathrm{g}j}-Ix_{i,j}). \tag{12}$$

In summary, employing Lévy flight in the ICOA offers benefits such as enhanced exploration, increased diversity, and improved convergence, making the algorithm more effective in solving complex optimization problems.

Remark 2 The reason for implementing the Lévy flight strategy with a 50% probability is to effectively balance exploration (global search) and exploitation (local search) abilities. Using Lévy flight exclusively might overly focus on a specific local area, reducing search diversity and flexibility. However, a 50% probability helps avoid local optima, maintains randomness, and enhances search efficiency and adaptability, increasing the likelihood of finding the global optimum.

3.3 Adaptive local search

In ICOA, the Phase 3 strategy is refined to enhance the exploitation phase, in particular improving the predator escape process. A significant advancement in this phase is the introduction of a dynamic scaling factor. Traditionally, COA adopts a relatively fixed method for updating positions during this phase. However, in ICOA, the updated position for each agent i and dimension j is calculated using the following dynamic approach:

$$\begin{aligned}x_{i,j}^{\mathrm{P2,\ new}}=x_{i,j}+s\cdot(1-2r)\cdot\big(&l_{\mathrm{b}j}^{\mathrm{local,new}}+\\&r\cdot(u_{\mathrm{b}j}^{\mathrm{local,new}}-l_{\mathrm{b}j}^{\mathrm{local,new}})\big),\end{aligned} \tag{13}$$

with

$$\begin{cases} l_{bj}^{\text{local, new}} = l_{bj} + \dfrac{u_{bj} - l_{bj}}{t}, & t = 1,\ 2,\ \cdots,\ T, \\ u_{bj}^{\text{local, new}} = u_{bj} - \dfrac{u_{bj} - l_{bj}}{t}, & t = 1,\ 2,\ \cdots,\ T. \end{cases} \tag{14}$$

The scale factor s is a dynamic parameter that changes with the iteration number t. It is calculated by

$$s = 0.1 + 0.3 \cdot \left(1 - \frac{t}{T}\right). \tag{15}$$

This dynamic scaling factor allows the algorithm to adaptively control the step size during the exploitation phase. At early iterations (small t) the scale factor is relatively large, enabling a more extensive search around the current position. As the iterations increase and t approaches T, the scale factor decreases, leading to a more refined search around promising regions.

Remark 3 This adaptive adjustment helps to balance exploration and exploitation throughout the optimization process. It makes the algorithm more efficient in finding better solutions by gradually focusing on the most promising areas, while maintaining some level of exploration to avoid getting stuck in local optima. Overall, this Phase 3 strategy provides a more intelligent and effective way of exploiting the search space compared to the simpler scheme in COA.

4 Simulations and analysis of results

4.1 Parameter settings

The simulation environment utilizes a 13th Generation Intel Core i5-13500H processor with 16 GB RAM on a Windows 10 64-bit system. To evaluate the effectiveness and superiority of the proposed ICOA, we conducted comparative simulation experiments with COA, PSO, ALO, GWO, and MFO algorithms. Unless otherwise specified, all subsequent simulations set the population size to N=30 and the maximum number of iterations to T=500. The parameter settings for each algorithm are detailed in Table 1.

Table 1 Parameter values for the compared algorithms

Algorithm	Parameter	Description
PSO	Topology	Fully connected
	Cognitive constant, C_1	2
	Social constant, C_2	2
	Inertia weight	Linearly decreasing from 0.9 to 0.6 (controling the impact of prior velocity)
	Maximum speed, v_{max}	1
MFO	Control gain	Linearly decreases from −1 to −2 (controls moth movement)
GWO	Control gain	Decreasing from 2 to 0 (controling wolf movement)

4.2 Description of benchmark functions

This study employs 12 benchmark functions for performance testing (Yao et al., 1999), as outlined in Table S1 of the electronic supplementary materials (ESM). The functions are categorized into two types: unimodal benchmark functions $\mathcal{F}_1$–$\mathcal{F}_7$ and multimodal benchmark functions $\mathcal{F}_8$–$\mathcal{F}_{12}$. The unimodal functions are utilized to assess the algorithm's development capability. In contrast, the multimodal functions contain both a global optimal solution and multiple local optimal solutions, thereby enabling evaluation of the algorithm's ability to perform global searches and escape local optima. By employing these two types of test functions, the optimization performance of the ICOA can be validated comprehensively.

4.3 Analysis of simulation results

To ensure the reliability of the experimental results, six algorithms are independently executed 30 times for each benchmark function. The convergence results and statistical results are presented in Figs. 2–4 and Table 2, respectively. It can be observed that the ICOA consistently achieved superior optimal convergence values and stability compared to the other five algorithms, for the majority of benchmark functions.

Specifically, for the unimodal benchmark functions $\mathcal{F}_1$–$\mathcal{F}_4$ and $\mathcal{F}_6$, ICOA successfully converges to the theoretical value of 0 with the fewest iterations required. This outcome indicates that ICOA possesses high convergence accuracy and convergence speed. Furthermore, in the experiments involving multimodal functions $\mathcal{F}_8$–$\mathcal{F}_{11}$, ICOA demonstrates excellent optimization accuracy and robustness, with consistent convergence to the theoretical optimal value observed in the experiments on $\mathcal{F}_9$ and $\mathcal{F}_{11}$.

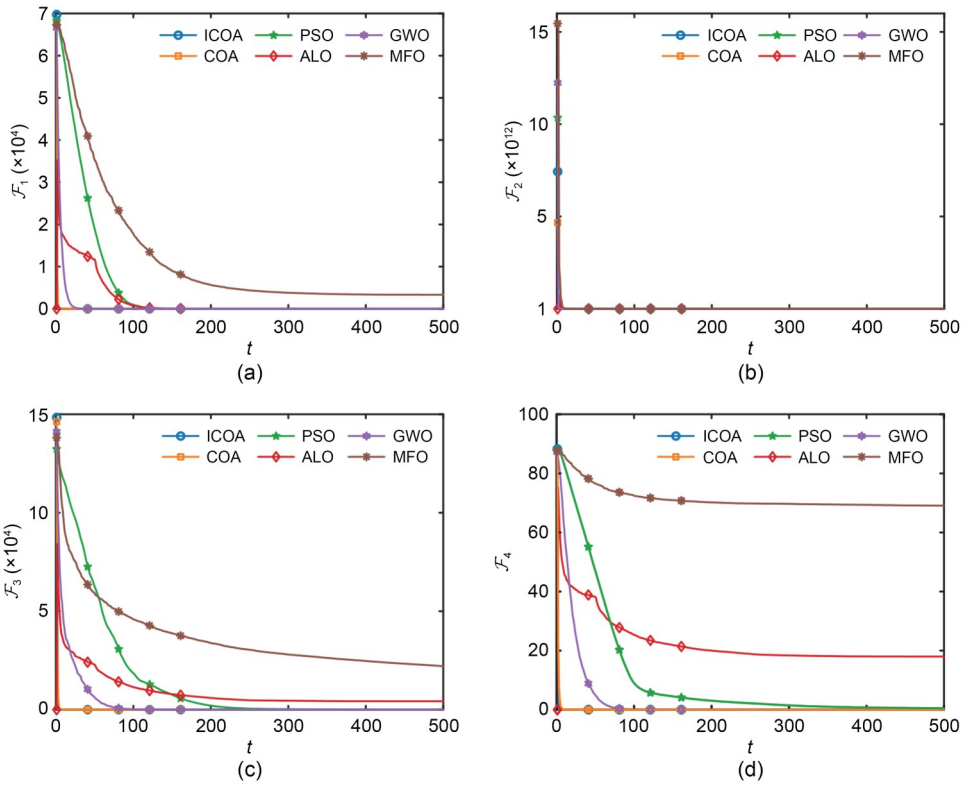

Fig. 2 Comparison of convergence results: (a) $\mathcal{F}_1$; (b) $\mathcal{F}_2$; (c) $\mathcal{F}_3$; (d) $\mathcal{F}_4$

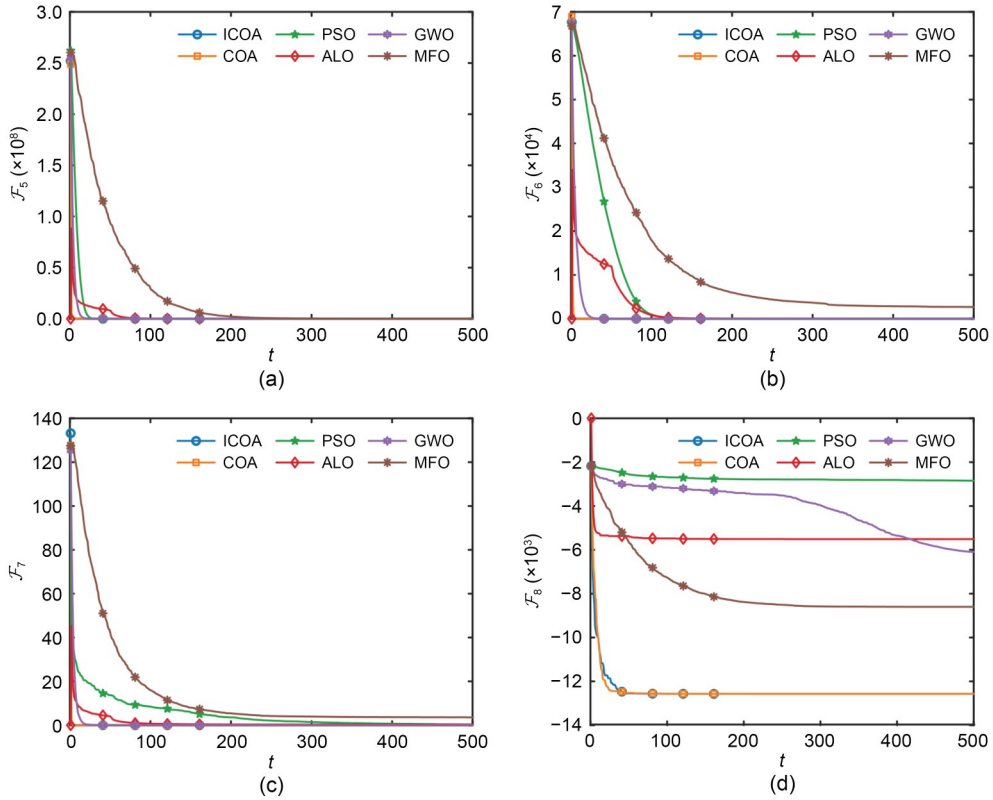

Fig. 3 Comparison of convergence results: (a) $\mathcal{F}_5$; (b) $\mathcal{F}_6$; (c) $\mathcal{F}_7$; (d) $\mathcal{F}_8$

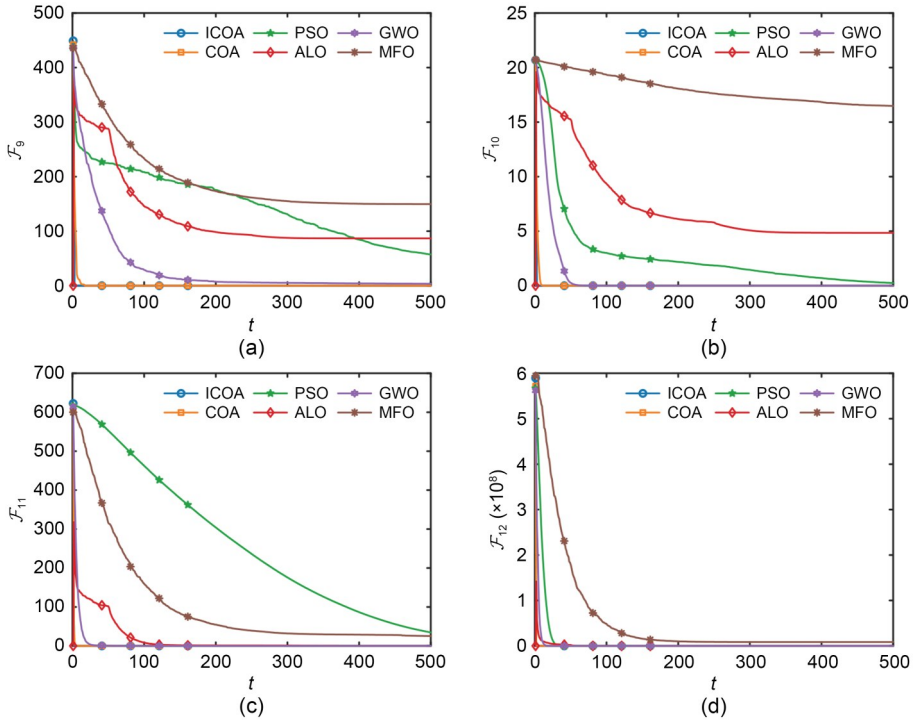

Fig. 4 Comparison of convergence results: (a) $\mathcal{F}_9$; (b) $\mathcal{F}_{10}$; (c) $\mathcal{F}_{11}$; (d) $\mathcal{F}_{12}$

Table 2 Statistics of the simulation results

Function	Index	ICOA	COA	PSO	ALO	GWO	MFO
$\mathcal{F}_1$	Mean	**0**	0	6.4981×10^{-2}	1.4891×10^{-3}	1.4849×10^{-7}	3.3369×10^{3}
	Std	**0**	0	2.8153×10^{-2}	1.4953×10^{-3}	1.9991×10^{-27}	6.0651×10^{3}
$\mathcal{F}_2$	Mean	**0**	5.0705×10^{-183}	7.4008×10^{-1}	6.1835×10^{1}	1.2026×10^{-16}	3.9691×10^{1}
	Std	**0**	0	2.1226×10^{-1}	5.0512×10^{1}	1.4013×10^{-16}	2.5103×10^{1}
$\mathcal{F}_3$	Mean	**0**	0	2.0881×10^{1}	4.2635×10^{3}	1.1973×10^{-5}	2.2044×10^{4}
	Std	**0**	0	7.6665	1.9577×10^{3}	2.8560×10^{-5}	1.1525×10^{4}
$\mathcal{F}_4$	Mean	**0**	7.4457×10^{-181}	4.8538×10^{-1}	1.7989×10^{1}	8.5723×10^{-7}	6.9062×10^{1}
	Std	**0**	0	3.2722×10^{-1}	5.8175	7.6335×10^{-7}	8.8306
$\mathcal{F}_5$	Mean	1.6510×10^{-3}	**0**	6.7068×10^{1}	3.7673×10^{2}	2.7076×10^{1}	1.0176×10^{4}
	Std	7.3731×10^{-3}	**0**	4.8673×10^{1}	5.7860×10^{2}	7.5212×10^{-1}	2.7262×10^{4}
$\mathcal{F}_6$	Mean	**0**	0	0	1.8300×10^{1}	0	2.6917×10^{3}
	Std	**0**	0	0	5.9199	0	5.2005×10^{3}
$\mathcal{F}_7$	Mean	$\mathbf{5.6370\times10^{-5}}$	5.2599×10^{-4}	3.3879×10^{-1}	2.6864×10^{-1}	1.8004×10^{-3}	3.5536
	Std	$\mathbf{4.9143\times10^{-5}}$	4.0657×10^{-4}	9.2339×10^{-2}	9.4353×10^{-2}	9.1275×10^{-4}	8.5897
$\mathcal{F}_8$	Mean	$\mathbf{-1.2569\times10^{4}}$	-1.2569×10^{4}	-2.8463×10^{3}	-5.5112×10^{3}	-6.0958×10^{3}	-8.6010×10^{3}
	Std	$\mathbf{1.0504\times10^{-1}}$	3.3467×10^{-1}	4.0004×10^{2}	3.3503×10^{2}	5.0699×10^{2}	9.3781×10^{2}
$\mathcal{F}_9$	Mean	**0**	0	5.7428×10^{1}	8.6729×10^{1}	3.5644	1.4958×10^{2}
	Std	**0**	0	1.3964×10^{1}	2.0242×10^{1}	4.7493	2.7364×10^{1}
$\mathcal{F}_{10}$	Mean	$\mathbf{8.8818\times10^{-16}}$	8.8818×10^{-16}	2.4598×10^{-1}	4.8471	1.0711×10^{-13}	1.6476×10^{1}
	Std	**0**	0	5.4681×10^{-2}	3.0114	1.9151×10^{-14}	6.2344
$\mathcal{F}_{11}$	Mean	**0**	0	3.4561×10^{1}	6.9750×10^{-2}	5.9996×10^{-3}	2.5359×10^{1}
	Std	**0**	0	4.7060	3.2149×10^{-2}	7.9950×10^{-3}	4.0276×10^{1}
$\mathcal{F}_{12}$	Mean	4.0754×10^{-7}	$\mathbf{1.5705\times10^{-32}}$	1.5894	1.4556×10^{1}	4.3799×10^{-2}	8.5340×10^{6}
	Std	1.8967×10^{-6}	$\mathbf{5.5674\times10^{-48}}$	1.2485	6.6168	2.1997×10^{-2}	4.6739×10^{7}

Bold data in Table 2 represent the best results

In summary, integrating LHS, Lévy flight, and adaptive local search significantly improves COA performance. These strategies collectively improve the initial population distribution, promoting a more diverse search space. This diversity is crucial for effective capturing of local optima, thereby preventing premature convergence. Consequently, the algorithm's convergence accuracy is enhanced, allowing it to reach the theoretical optimum with fewer iterations.

4.4 Wilcoxon rank sum test

To further investigate the differences between ICOA and other swarm intelligence algorithms, a statistical analysis is performed. Specifically, the Wilcoxon rank sum test is applied to the test set with a significance level of 5%. A *P*-value of less than 5% indicates a significant difference between the two algorithms for a given test function. Conversely, a result of NaN signifies that the optimization performance of the two algorithms is equivalent (Hussien et al., 2020; Saheed et al., 2023).

Table 3 presents the statistical results of the Wilcoxon rank sum test *P*-values for ICOA and the algorithms COA, PSO, ALO, GWO, and MFO, under the condition of a 30-dimensional space and 30 independent times. The symbols "+/−/NaN" are used to summarize the simulation results in Table 2, representing "excellent/poor/equivalent" performance, respectively. As indicated in Table 3, significant differences exist between ICOA and the five aforementioned algorithms. For all benchmark functions except $\mathcal{F}_6$ (where the *P*-value for COA is 0.057738, slightly above the 5% significance threshold), the *P*-values are below 0.05, confirming statistically significant differences in performance. Additionally, ICOA achieves better performance statistically in 11 out of 12 benchmark functions compared to COA, and in all 12 benchmark functions compared to PSO, ALO, GWO, and MFO. These results demonstrate the consistent superiority of ICOA across a wide range of benchmark functions.

5 Comparative analysis of engineering applications

Tension/compression springs are employed in a variety of engineering applications owing to their multifaceted functionality. A prime example can be seen in the trigger systems of firearms. Here, compressional springs store and release energy upon the shooter pulling the trigger, thus controlling the trigger's travel and providing force feedback. The design of tension/compression springs is a nonlinear optimization problem marked by multiple design variables and complex practical constraints. This complexity poses significant challenges to swarm intelligence optimization algorithms, and makes spring design a good test for evaluating the optimization capabilities of such algorithms.

Based on Arora (2004), the tension/compression spring design problem aims to minimize the weight of

Table 3 Wilcoxon rank sum test *P*-values for different algorithms

Function	Wilcoxon rank sum test *P*-value				
	COA	PSO	ALO	GWO	MFO
$\mathcal{F}_1$	2.6979×10^{-156}	3.7554×10^{-185}	1.5556×10^{-184}	5.5692×10^{-185}	1.9687×10^{-185}
$\mathcal{F}_2$	5.9203×10^{-185}	5.7613×10^{-185}	2.1249×10^{-184}	5.8009×10^{-185}	4.9618×10^{-185}
$\mathcal{F}_3$	3.5954×10^{-153}	3.9923×10^{-185}	2.2138×10^{-184}	5.6454×10^{-185}	3.5087×10^{-185}
$\mathcal{F}_4$	5.8802×10^{-185}	3.4613×10^{-185}	6.5667×10^{-185}	4.9619×10^{-185}	1.9836×10^{-186}
$\mathcal{F}_5$	2.9549×10^{-143}	2.0209×10^{-162}	3.2441×10^{-161}	2.7635×10^{-161}	7.8826×10^{-163}
$\mathcal{F}_6$	5.7738×10^{-2}	6.9653×10^{-154}	4.7857×10^{-185}	3.9612×10^{-17}	1.8899×10^{-185}
$\mathcal{F}_7$	1.6947×10^{-100}	5.6663×10^{-163}	3.0597×10^{-161}	2.5785×10^{-133}	1.1033×10^{-163}
$\mathcal{F}_8$	1.8056×10^{-50}	1.1491×10^{-163}	1.0505×10^{-162}	1.9135×10^{-163}	8.0966×10^{-160}
$\mathcal{F}_9$	7.4359×10^{-9}	5.7986×10^{-185}	1.6091×10^{-184}	5.4941×10^{-185}	3.8068×10^{-185}
$\mathcal{F}_{10}$	1.0299×10^{-11}	4.9955×10^{-185}	1.1458×10^{-183}	4.1970×10^{-185}	1.9568×10^{-186}
$\mathcal{F}_{11}$	1.8977×10^{-7}	3.1536×10^{-186}	1.5451×10^{-184}	5.5266×10^{-185}	1.7899×10^{-185}
$\mathcal{F}_{12}$	2.4572×10^{-117}	2.0006×10^{-162}	3.3082×10^{-161}	3.2884×10^{-161}	1.1411×10^{-163}
+/−/NaN	11/1/0	12/0/0	12/0/0	12/0/0	12/0/0

the spring. As depicted in Fig. 5, this problem involves three continuous variables: wire diameter (d), average coil diameter (D), and the number of active coils (P) (Altay et al., 2024). The mathematical model is presented in the ESM.

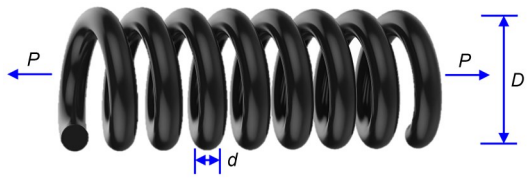

Fig. 5 Tension/compression spring design

Consequently, this section focuses on assessing the applicability and advantages of ICOA through testing on the tension/compression spring design problem. To reach the optimization goal, ICOA's performance is compared with seven other swarm intelligence algorithms. Here, SCA means sinusoidal coordinate algorithm, GSA means gravitational search algorithm, and HHO means Harris hawks optimization.

It is important to note that the simulation results for the other algorithms are derived from values reported in existing literature. In contrast, the ICOA is executed independently 30 times, with each execution allowing a maximum of 1000 iterations to determine the optimal result. The results of these simulations are presented in Fig. 6 and Table 4.

The simulation results demonstrate that ICOA exhibits superior optimization performance on a tension/compression spring design problem compared to seven other swarm intelligence algorithms, thereby confirming the effectiveness of the improvement strategies described in this study.

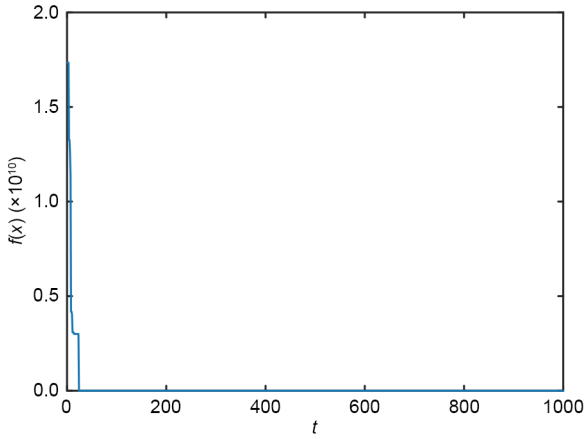

Fig. 6 Convergence process of ICOA applied to the tension/compression spring design problem

6 Conclusions

This study introduces an ICOA that addresses the limitation of the traditional COA, such as limited population diversity and a tendency to become trapped in local optima. The main conclusions are as follows.

(1) The integration of LHS, Lévy-flight distribution, and adaptive local search strategies significantly enhanced both the quality and diversity of the population, thereby improving the algorithm's performance. LHS offers a robust starting point for the search process, ensuring a well-distributed initial population. Meanwhile, Lévy-flight introduces an efficient random walk mechanism that facilitates global exploration. The adaptive local search strategy complements both these strategies by fine-tuning solutions for better exploitation. This synergistic combination leads to

Table 4 Optimization results for the tension/compression spring design problem

Algorithm	Study	d (m)	D (m)	P	Optimal $f(x)$
ICOA	This paper	5.1706×10^{-2}	3.5713×10^{-1}	1.1265×10^{1}	**1.2665×10^{-2}**
COA	Dehghani et al. (2023)	5.1913×10^{-2}	3.6212×10^{-1}	1.0979×10^{1}	1.2666×10^{-2}
PSO	He and Wang (2007)	5.1728×10^{-2}	3.5764×10^{-1}	1.1245×10^{1}	1.2675×10^{-2}
SCA	Braik (2021)	5.0780×10^{-2}	3.3478×10^{-1}	1.2723×10^{1}	1.2710×10^{-2}
GWO	Mirjalili et al. (2014)	5.1690×10^{-2}	3.5674×10^{-1}	1.1289×10^{1}	1.2666×10^{-2}
MFO	Mirjalili (2015b)	5.1994×10^{-2}	3.6411×10^{-1}	1.0868×10^{1}	1.2667×10^{-2}
GSA	Houssein et al. (2021)	5.3903×10^{-2}	4.0698×10^{-1}	9.3675	1.3442×10^{-2}
HHO	Houssein et al. (2021)	5.1416×10^{-2}	3.5018×10^{-1}	1.1683×10^{1}	1.2667×10^{-2}

Bold data in Table 4 represents the best result

improved optimization ability and convergence accuracy for the algorithm.

(2) The proposed ICOA was tested using 12 benchmark functions, with its performance compared to traditional optimization algorithms including COA, PSO, ALO, GWO, and MFO. Analysis of convergence, stability, and statistical significance was performed, as assessed through the Wilcoxon rank sum test. The simulation results showcased the superior optimization capabilities of the ICOA algorithm. Its improved balance between exploration and exploitation enabled better identification of global optima across complex fitness landscapes.

(3) The practical applicability and efficacy of the ICOA have been substantiated through successful application on design optimization of tension/compression springs. This real-world case study validates the theoretical advantages of ICOA, and underscores its potential to address complex engineering optimization challenges with enhanced efficiency and reliability.

Future research will aim to enhance the performance of ICOA by broadening its application to a wider range of defense engineering optimization problems. Furthermore, we will investigate the integration of machine learning techniques to predict and guide the search process, with the goal of developing a more intelligent and adaptive optimization framework.

Acknowledgments

This work is supported by the Natural Science Foundation of Hunan Province of China (Nos. 2021JJ10045 and 2025JJ60072), the Open Research Subject of State Key Laboratory of Intelligent Game (No. ZBKF-24-01), the Postdoctoral Fellowship Program of CPSF (No. GZB20240989), and the China Postdoctoral Science Foundation (No. 2024M754304).

Author contributions

Shuangxi LIU designed the research and wrote the first draft of the manuscript. Ruizhe FENG and Yuxin WEI processed the corresponding data and helped to organize the manuscript. Wei HUANG and Binbin YAN revised and edited the final version.

Conflict of interest

Wei HUANG is an Editorial Board member of this journal, and is NOT involved in the editorial review or the decision to publish this article. Shuangxi LIU, Ruizhe FENG, Yuxin WEI, Wei HUANG, and Binbin YAN declare that they have no conflict of interest.

Data availability statement

The data, models, and code used in this study are proprietary or confidential in nature. Due to restrictions, the codes cannot be made publicly available.

References

Agushaka JO, Ezugwu AE, Abualigah L, 2023. Gazelle optimization algorithm: a novel nature-inspired metaheuristic optimizer. *Neural Computing and Applications*, 35(5): 4099-4131. https://doi.org/10.1007/s00521-022-07854-6

Ali ES, Abd Elazim SM, Abdelaziz AY, 2017. Ant lion optimization algorithm for optimal location and sizing of renewable distributed generations. *Renewable Energy*, 101: 1311-1324. https://doi.org/10.1016/j.renene.2016.09.023

Altay EV, Altay O, Özçevik Y, 2024. A comparative study of metaheuristic optimization algorithms for solving real-world engineering design problems. *Computer Modeling in Engineering & Sciences*, 139(1):1039-1094. https://doi.org/10.32604/cmes.2023.029404

Arora JS, 2004. Introduction to Optimum Design. 2nd Edition. Elsevier, Amsterdam, the Netherlands. https://doi.org/10.1016/B978-0-12-064155-0.X5000-9

Bingul Z, Karahan O, 2018. A novel performance criterion approach to optimum design of PID controller using cuckoo search algorithm for AVR system. *Journal of the Franklin Institute*, 355(13):5534-5559. https://doi.org/10.1016/j.jfranklin.2018.05.056

Braik MS, 2021. Chameleon swarm algorithm: a bio-inspired optimizer for solving engineering design problems. *Expert Systems with Applications*, 174:114685. https://doi.org/10.1016/j.eswa.2021.114685

Brezočnik L, Fister Jr I, Podgorelec V, 2018. Swarm intelligence algorithms for feature selection: a review. *Applied Sciences*, 8(9):1521. https://doi.org/10.3390/app8091521

Cui ZH, Zhang ZX, Hu ZM, et al., 2022. A many-objective optimization based intelligent high performance data processing model for cyber-physical-social systems. *IEEE Transactions on Network Science and Engineering*, 9(6):3825-3834. https://doi.org/10.1109/TNSE.2021.3073911

Cui ZL, Li CQ, Huang JR, et al., 2020. An improved moth flame optimization algorithm for minimizing specific fuel consumption of variable cycle engine. *IEEE Access*, 8: 142725-142735. https://doi.org/10.1109/ACCESS.2020.3001156

Dehghani M, Montazeri Z, Trojovská E, et al., 2023. Coati optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems. *Knowledge-Based Systems*, 259:110011. https://doi.org/10.1016/j.knosys.2022.110011

Emam MM, Houssein EH, Samee NA, et al., 2024. Breast cancer diagnosis using optimized deep convolutional neural network based on transfer learning technique and improved

coati optimization algorithm. *Expert Systems with Applications*, 255:124581. https://doi.org/10.1016/j.eswa.2024.124581

Emary E, Zawbaa HM, Sharawi M, 2019. Impact of Lèvy flight on modern meta-heuristic optimizers. *Applied Soft Computing*, 75:775-789. https://doi.org/10.1016/j.asoc.2018.11.033

Faris H, Aljarah I, Al-Betar MA, et al., 2018. Grey wolf optimizer: a review of recent variants and applications. *Neural Computing and Applications*, 30(2):413-435. https://doi.org/10.1007/s00521-017-3272-5

Hashim FA, Hussain K, Houssein EH, et al., 2021. Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. *Applied Intelligence*, 51(3): 1531-1551. https://doi.org/10.1007/s10489-020-01893-z

Hashim FA, Houssein EH, Mostafa RR, et al., 2023. An efficient adaptive-mutated coati optimization algorithm for feature selection and global optimization. *Alexandria Engineering Journal*, 85:29-48. https://doi.org/10.1016/j.aej.2023.11.004

He Q, Wang L, 2007. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence*, 20(1):89-99. https://doi.org/10.1016/j.engappai.2006.03.003

He QQ, Liu H, Ding GY, et al., 2023. A modified Lévy flight distribution for solving high-dimensional numerical optimization problems. *Mathematics and Computers in Simulation*, 204:376-400. https://doi.org/10.1016/j.matcom.2022.08.017

Heise SA, Morse HS, 2000. The DARPA JFACC program: modeling and control of military operations. Proceedings of the 39th IEEE Conference on Decision and Control, p.2551-2555. https://doi.org/10.1109/CDC.2000.914187

Houssein EH, Saad MR, Hashim FA, et al., 2020. Lévy flight distribution: a new metaheuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 94:103731. https://doi.org/10.1016/j.engappai.2020.103731

Houssein EH, Mahdy MA, Blondin MJ, et al., 2021. Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems. *Expert Systems with Applications*, 174: 114689. https://doi.org/10.1016/j.eswa.2021.114689

Hussien AG, Hassanien AE, Houssein EH, et al., 2020. New binary whale optimization algorithm for discrete optimization problems. *Engineering Optimization*, 52(6):945-959. https://doi.org/10.1080/0305215X.2019.1624740

Iacca G, dos Santos Junior VC, de Melo VV, 2021. An improved Jaya optimization algorithm with Lévy flight. *Expert Systems with Applications*, 165:113902. https://doi.org/10.1016/j.eswa.2020.113902

Karimi J, Rajabi MR, Sadati SH, et al., 2024. Multidisciplinary design optimization of a dual-spin guided vehicle. *Defence Technology*, 37:133-148. https://doi.org/10.1016/j.dt.2023.11.025

Kennedy J, Eberhart R, 1995. Particle swarm optimization. Proceedings of International Conference on Neural Networks, p.1942-1948. https://doi.org/10.1109/ICNN.1995.488968

Le THH, Dinh PH, Vu VH, et al., 2024. A new approach to medical image fusion based on the improved extended difference-of-Gaussians combined with the coati optimization algorithm. *Biomedical Signal Processing and Control*, 93:106175. https://doi.org/10.1016/j.bspc.2024.106175

Li BW, Ye SX, Qi Liang, et al., 2024. Data correction method for low-cost gas sensors based on COA-GRU algorithm. *Instrument Technique and Sensor*, (3):120-126 (in Chinese). https://doi.org/10.3969/j.issn.1002-1841.2024.03.022

Li G, Yao Y, Shen LJ, et al., 2023. Influence of yaw damper layouts on locomotive lateral dynamics performance: Pareto optimization and parameter analysis. *Journal of Zhejiang University-SCIENCE A*, 24(5):450-464. https://doi.org/10.1631/jzus.A2200374

Liu SX, Huang FP, Yan BB, et al., 2021. Optimal design of multimissile formation based on an adaptive SA-PSO algorithm. *Aerospace*, 9(1):21. https://doi.org/10.3390/aerospace9010021

Liu SX, Liu W, Huang F, et al., 2022. Multitarget allocation strategy based on adaptive SA-PSO algorithm. *The Aeronautical Journal*, 126(1300):1069-1081. https://doi.org/10.1017/aer.2021.124

Liu SX, Lin ZH, Huang W, et al., 2025. Current development and future prospects of multi-target assignment problem: a bibliometric analysis review. *Defence Technology*, 43: 44-59. https://doi.org/10.1016/j.dt.2024.09.006

Liu XY, Li GQ, Yang HY, et al., 2023. Agricultural UAV trajectory planning by incorporating multi-mechanism improved grey wolf optimization algorithm. *Expert Systems with Applications*, 233:120946. https://doi.org/10.1016/j.eswa.2023.120946

Ma WH, Fang YW, Fu WX, et al., 2023. Cooperative localisation of UAV swarm based on adaptive SA-PSO algorithm. *The Aeronautical Journal*, 127(1307):57-75. https://doi.org/10.1017/aer.2022.54

Martí Sempere C, 2023. The problem of allocating resources to defense. *Defence Studies*, 23(1):86-104. https://doi.org/10.1080/14702436.2022.2094251

Mirjalili S, Mirjalili SM, Lewis A, 2014. Grey wolf optimizer. *Advances in Engineering Software*, 69:46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Mirjalili S, 2015a. The ant lion optimizer. *Advances in Engineering Software*, 83:80-98. https://doi.org/10.1016/j.advengsoft.2015.01.010

Mirjalili S, 2015b. Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 89:228-249. https://doi.org/10.1016/j.knosys.2015.07.006

Pestana Barros C, 2004. Measuring performance in defense-sector companies in a small NATO member country. *Journal of Economic Studies*, 31(2):112-128. https://doi.org/10.1108/01443580410527105

Politis SS, Zhang ZM, Han Z, et al., 2021. Stochastic analysis of network-level bridge maintenance needs using Latin hypercube sampling. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, 7(1):04020049. https://doi.org/10.1061/AJRUA6.0001099

Rajabi MM, Ataie-Ashtiani B, Janssen H, 2015. Efficiency enhancement of optimized Latin hypercube sampling strategies: application to Monte Carlo uncertainty analysis and meta-modeling. *Advances in Water Resources*, 76:127-139. https://doi.org/10.1016/j.advwatres.2014.12.008

Saeed RA, Omri M, Abdel-Khalek S, et al., 2022. Optimal path planning for drones based on swarm intelligence algorithm. *Neural Computing and Applications*, 34(12):10133-10155. https://doi.org/10.1007/s00521-022-06998-9

Saheed YK, Balogun BF, Odunayo BJ, et al., 2023. Microarray gene expression data classification via Wilcoxon sign rank sum and novel grey wolf optimized ensemble learning models. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(6):3575-3587. https://doi.org/10.1109/TCBB.2023.3305429

Sahoo SK, Saha AK, 2022. A hybrid moth flame optimization algorithm for global optimization. *Journal of Bionic Engineering*, 19(5):1522-1543. https://doi.org/10.1007/s42235-022-00207-y

Shamami N, Mehdizadeh E, Yazdani M, et al., 2024. War game problem considering the mobility of weapons and targets. *Journal of Engineering Research*, 12(1):214-225. https://doi.org/10.1016/j.jer.2023.11.021

Sharma A, Shoval S, Sharma A, et al., 2022. Path planning for multiple targets interception by the swarm of UAVs based on swarm intelligence algorithms: a review. *IETE Technical Review*, 39(3):675-697. https://doi.org/10.1080/02564602.2021.1894250

Shields MD, Zhang JX, 2016. The generalization of Latin hypercube sampling. *Reliability Engineering & System Safety*, 148:96-108. https://doi.org/10.1016/j.ress.2015.12.002

Song YY, Wang FL, Chen XX, 2019. An improved genetic algorithm for numerical function optimization. *Applied Intelligence*, 49(5):1880-1902. https://doi.org/10.1007/s10489-018-1370-4

Sun WC, Lin SD, Zhang H, et al., 2024. A reduced combustion mechanism of ammonia/diesel optimized with multi-objective genetic algorithm. *Defence Technology*, 34:187-200. https://doi.org/10.1016/j.dt.2023.11.008

Terziev V, Nichev N, 2017. Streamlining management solutions for economic, effective and efficient spending of resources for security and defense. *IJASOS-International E-Journal of Advances in Social Sciences*, 3(8):640-644. https://doi.org/10.18769/ijasos.337320

Wade BM, 2019. A multi-objective optimization of ballistic and cruise missile fire plans based on damage calculations from missile impacts on an airfield defended by an air defense artillery network. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 16(2):103-117. https://doi.org/10.1177/1548512918788503

Wang CG, Yan JH, Li WL, et al., 2024. Disturbances rejection optimization based on improved two-degree-of-freedom LADRC for permanent magnet synchronous motor systems. *Defence Technology*, 33:518-531. https://doi.org/10.1016/j.dt.2023.11.007

Wang DS, Tan DP, Liu L, 2018. Particle swarm optimization algorithm: an overview. *Soft Computing*, 22(2):387-408. https://doi.org/10.1007/s00500-016-2474-6

Wang Y, Gao HL, Bao XY, et al., 2024. Load frequency control of power system based on coati optimization algorithm. *Automation & Instrumentation*, 39(9):37-40 (in Chinese). https://doi.org/10.19557/j.cnki.1001-9944.2024.09.009

Yang ZP, Yang SN, Zhou QS, et al., 2022. A joint optimization algorithm for focused energy delivery in precision electronic warfare. *Defence Technology*, 18(4):709-721. https://doi.org/10.1016/j.dt.2021.03.001

Yao P, Wang HL, 2017. Dynamic adaptive ant lion optimizer applied to route planning for unmanned aerial vehicle. *Soft Computing*, 21(18):5475-5488. https://doi.org/10.1007/s00500-016-2138-6

Yao X, Liu Y, Lin GM, 1999. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3(2):82-102. https://doi.org/10.1109/4235.771163

Ye XW, Zhang XL, Chen YB, et al., 2024. Prediction of maximum upward displacement of shield tunnel linings during construction using particle swarm optimization-random forest algorithm. *Journal of Zhejiang University-SCIENCE A*, 25(1):1-17. https://doi.org/10.1631/jzus.A2300011

Zhang F, Cheng L, Wu MY, et al., 2020. Performance analysis of two-stage thermoelectric generator model based on Latin hypercube sampling. *Energy Conversion and Management*, 221:113159. https://doi.org/10.1016/j.enconman.2020.113159

Zhao TH, Wu LJ, Cui ZH, et al., 2025. An adaptive strategy based multi-population multi-objective optimization algorithm. *Information Sciences*, 686:120913. https://doi.org/10.1016/j.ins.2024.120913

Zhao ZQ, Li T, Sheng DL, et al., 2024. Machine learning optimization strategy of shaped charge liner structure based on jet penetration efficiency. *Defence Technology*, 39:23-41. https://doi.org/10.1016/j.dt.2024.04.006

Electronic supplementary materials

Sections S1 and S2