Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.049717

ARTICLE

A Multi-Strategy-Improved Northern Goshawk Optimization Algorithm for Global Optimization and Engineering Design

Liang Zeng[1,2], Mai Hu[1], Chenning Zhang[1], Quan Yuan[1] and Shanshan Wang[1,2,*]

[1]School of Electrical and Electronic Engineering, Hubei University of Technology, Wuhan, 430068, China

[2]Hubei Key Laboratory for High-Efficiency Utilization of Solar Energy and Operation Control of Energy Storage System, Hubei University of Technology, Wuhan, 430068, China

*Corresponding Author: Shanshan Wang. Email: wangshanshan@hbut.edu.cn

Received: 16 January 2024 Accepted: 11 April 2024 Published: 18 July 2024

ABSTRACT

Optimization algorithms play a pivotal role in enhancing the performance and efficiency of systems across various scientific and engineering disciplines. To enhance the performance and alleviate the limitations of the Northern Goshawk Optimization (NGO) algorithm, particularly its tendency towards premature convergence and entrapment in local optima during function optimization processes, this study introduces an advanced Improved Northern Goshawk Optimization (INGO) algorithm. This algorithm incorporates a multifaceted enhancement strategy to boost operational efficiency. Initially, a tent chaotic map is employed in the initialization phase to generate a diverse initial population, providing high-quality feasible solutions. Subsequently, after the first phase of the NGO's iterative process, a whale fall strategy is introduced to prevent premature convergence into local optima. This is followed by the integration of T-distribution mutation strategies and the State Transition Algorithm (STA) after the second phase of the NGO, achieving a balanced synergy between the algorithm's exploitation and exploration. This research evaluates the performance of INGO using 23 benchmark functions alongside the IEEE CEC 2017 benchmark functions, accompanied by a statistical analysis of the results. The experimental outcomes demonstrate INGO's superior achievements in function optimization tasks. Furthermore, its applicability in solving engineering design problems was verified through simulations on Unmanned Aerial Vehicle (UAV) trajectory planning issues, establishing INGO's capability in addressing complex optimization challenges.

KEYWORDS

Northern Goshawk Optimization; tent chaotic map; T-distribution disturbance; state transition algorithm; UAV path planning

1 Introduction

The growing complexity of optimization problems, combined with the limitations of traditional algorithms, has underscored the need for developing meta-heuristic algorithms in recent years [1]. Focusing solely on inputs and outputs, metaheuristic algorithms excel in solving a broad spectrum of complex optimization problems across different fields. Applications range from path planning [2], and scheduling [3,4], to image threshold segmentation [5], and feature extraction and selection [6].

The inspiration behind metaheuristic algorithms stems from natural phenomena such as evolution, fundamental physicochemical laws, and the intelligent behaviors of swarms. By simulating these behaviors to construct models, these algorithms approximate the global optima of nonlinear, non-convex, discontinuous, and discrete complex optimization problems, thus achieving high-quality solutions at lower computational costs.

Metaheuristic algorithms are categorized into four types: Evolutionary, swarm intelligence-based, physicochemical law-grounded, and social or human behavior-modeled. The Genetic Algorithm (GA) [7] is notably the most influential among evolution-based algorithms. is notably the most influential among evolution-based algorithms. Mimicking Darwin's evolution process, GA assesses organism fitness to their environments and uses genetic operators—selection, crossover, mutation—to iteratively improve solution quality, effectively solving optimization problems. Evolutionary algorithms, applicable broadly and independent of gradient information, are marked by high computational costs and slow convergence. Swarm intelligence algorithms replicate group information exchange capabilities and characteristics. For example, the particle swarm optimization (PSO) [8] mirrors birds' predation behavior, simplifying optimization problem-solving to a quest for nourishment. Leveraging population and evolution concepts, these algorithms facilitate complex solution space exploration via individual cooperation. A common trait of swarm intelligence algorithms is their vulnerability to local optima, resulting in quicker convergence at the risk of settling for suboptimal solutions. Algorithms grounded in physical and chemical principles, like the Simulated Annealing algorithm (SA) [9] and Sine and Cosine Algorithm (SCA) [10], offer superior convergence accuracy. However, these algorithms have notable drawbacks, such as an excessive dependency on parameter settings, slow convergence rates, and applicability to only specific types of optimization problems. Algorithms inspired by social or human behaviors, like the Brain Storm Optimization Algorithm (BSOA) [11] and Tabu Search (TS) [12], derive their search strategies from the simulation of human behaviors in social interactions, learning, competition, and cooperation, aiming to solve complex optimization problems. The design of their search strategies significantly influences the convergence speed and stability of these algorithms.

Swarm intelligence algorithms offer significant benefits and potential for solving real-world engineering problems. As a result, numerous novel and enhanced swarm intelligence algorithms have been developed to address practical challenges. Introduced in the early 1990s, the Ant Colony Algorithm (ACO) [13] mimics ants' foraging behavior to find the shortest path. It uses stochastic state transitions and pheromone updates, effectively solving the Traveling Salesman Problem. After 2000, swarm intelligence algorithms gained researchers' attention for their fast convergence and robust stability. The Grey Wolf Optimizer (GWO) [14] was proposed in 2014, followed by the Whale Optimization Algorithm (WOA) [15] in 2016, igniting a surge in swarm intelligence algorithm proposals and improvements. In recent years, algorithms including the Dragonfly Algorithm (DA) [16], Aquila Optimizer (AO) [17], Harris Hawks Optimization (HHO) [18], Hunger Games Search (HGS) [19], Prairie Dog Optimization Algorithm (PDO) [20], Dwarf Mongoose Optimization Algorithm (DMO) [21], Chimp Optimization Algorithm (ChOA) [22], Beluga Whale Optimization (BWO) [23], have been introduced and shown effectiveness in various benchmarks and specific problems. The introduction of these algorithms has markedly facilitated solving complex engineering challenges in the real world.

Dehghani et al. [24] proposed the NGO in 2022, building on the simulation of the Northern goshawk's predation behavior, featuring a mathematically simple model and rapid computation speeds, finding application across numerous domains. Yu et al. [25] enhanced NGO's performance through methods such as leader mutation and sine chaos mapping, proposing a Chaotic Leadership Northern Goshawk Optimization (CLNGO), effectively addressing rolling bearing fault diagnosis issues. Youssef et al. [26] introduced an Enhanced Northern Goshawk Optimization (ENGO) for smart home energy management, incorporating Brownian motion in the first phase and Levy flight strategy in the second phase of NGO, which, although improving global search capability to some extent, rendered the algorithm prone to convergence difficulties. Liang et al. [27] improved NGO by integrating a polynomial interpolation strategy and multi-strategy opposite learning method, evaluating its performance using multiple benchmark function test sets, achieving noteworthy results, albeit the search methodologies of these strategies were relatively singular compared to others. El-Dabah et al. [28] employed NGO for identifying Photovoltaic (PV) model parameters. Satria et al. [29] introduced a hybrid algorithm, Chaotic Northern Goshawk, and Pattern Search (CNGPS), merging NGO with pattern search to improve convergence accuracy in PV parameter identification, reducing early-stage local optima's impact and preventing premature convergence. Zhong et al. [30] optimized the hyperparameters of the Bi-directional Long Short Term Memory (BILSTM) network for facial expression recognition using NGO, with the proposed NGO-BILSTM model achieving an accuracy of 89.72%. Tian et al. [31] extended NGO into a multi-objective version to address multi-objective Disassembly Line Balancing Problems (DLBP) in remanufacturing, significantly contributing to solving DLBP issues. Wang et al. [32] refined NGO using Tent chaotic mapping, adaptive weight factor, and Levy flight strategy, applying the novel algorithm to tool wear condition identification, reaching an accuracy of 97.9%. Zhan et al. [33] expanded NGO into Multi-objective Northern Goshawk Optimization Algorithm (MONGO) for environmentally oriented disassembly planning of end-of-life vehicle batteries, aiming to maximize disassembly efficiency and quality. Bian et al. [34] combined NGO with Henry Gas Solubility Optimization (HGSO) to create the eXtreme Gradient Boosting Northern Goshawk (XGBNG) model, which predicts the compressive strength of self-compacting concrete with unparalleled accuracy. Wang et al. [35] applied NGO to address buyer-seller behavior models in the electric power trading system, effectively enhancing the rational reutilization of electricity in private electric vehicles. Ning et al. [36] proposed an Artificial Neural Network (ANN) model optimized by NGO, resulting in the NGO-ANN model with high predictive accuracy.

Despite the emergence of numerous NGO variants, they exhibit limitations and excel only in specific problems. This reality persists regardless of the type or extent of modifications researchers apply to any algorithm, as each harbors its inherent constraints. The No Free Lunch (NFL) theorem [37] elucidates that for any optimization algorithm, the expected performance across all possible problems is equivalent, signifying that no algorithm can universally attain the global optimum for every problem. This theorem has led researchers to work on improving existing algorithms and introducing new ones and their enhanced versions. Based on this principle, it is imperative to continuously propose new algorithms and adapt these proposals to address the characteristics of different types of problems to tackle both current and forthcoming optimization challenges [38]. NGO exhibits exceptional performance across a variety of optimization problems; however, it is undeniable that limitations remain. The research improvements on NGO have not transcended beyond optimization initialization, parameter enhancement, and mutation strategies. These strategies effectively ameliorate the algorithm's convergence speed and, to some extent, its global search capability, yet their impact on the precision of NGO's convergence is minimal. This article, starting from the NFL theorem and focusing on the minimization of functions as the problem type, addresses the issue of suboptimal convergence accuracy inherent in both NGO and its enhancements. Innovatively, this work integrates STA [39] into NGO, proposing a multi-strategy improved version, INGO, aimed at augmenting the algorithm's local exploitation capacity. The performance of NGO is evaluated on two public test sets and one engineering problem. The contribution of this study is not limited to presenting an optimization algorithm with superior performance but also lies in the in-depth exploration of strategies for algorithm improvement, offering new insights and methodologies for future related research.

Furthermore, through the application case of UAV path planning, the potential of INGO in practical engineering problems is showcased, further demonstrating its value as an effective tool for engineering design and global optimization. Following is a summary of the paper's main contributions:

(1) This study synthesizes the high convergence accuracy of physics-based algorithms and the rapid convergence speed of swarm intelligence algorithms by incorporating the STA into the optimization process of NGO, thereby proposing a novel algorithm with superior performance, denoted as INGO.
(2) In the initialization phase, INGO utilizes Tent chaotic mapping to seed the population, thereby augmenting population diversity and boosting the chances of circumventing local optima in pursuit of superior solutions.
(3) The integration of the BWO's whale fall strategy and the T-distribution variation strategy as mutation strategies significantly augments the global search capability of NGO.
(4) The incorporation of the STA facilitates sampling and comparison within the vicinity of the optimal solution in each iteration of the NGO, thereby enhancing the precision of solution convergence.
(5) The performance of INGO was tested and evaluated using 53 benchmark functions and a path planning scenario of UAV.

The structure of this paper is as follows: Section 2 delineates the foundational principles and models of NGO. Section 3 outlines the development of the INGO algorithm. Section 4 reports on the evaluation of the algorithm's efficacy through testing on 53 benchmark functions. Section 5 elaborates on the deployment of this algorithm in UAV path planning and shares the outcomes. Section 6 offers a concluding summary of the paper.

2 Northern Goshawk Optimization (NGO)

Inspired by the predatory behavior of the northern goshawk, NGO takes each northern goshawk as an individual of a population and abstracts the predatory behavior into two stages. The first stage is to explore and attack prey, which reflects the group behavior by identifying the position of the current best solution and influencing each individual to update their position through the optimal solution, representing the global exploration ability. The second stage is the escape behavior of prey, symbolizing the local exploitation ability.

2.1 Initialization of the Algorithm

NGO considers each northern goshawk as an individual of a population, that is, a viable solution, and searches for the best value in the viable solution space by moving. Like most swarm intelligence algorithms, NGO generates initial populations by randomly generating initial populations. In the mathematical model, each individual represents a D-dimensional vector, N individuals constitute the entire population, and the population is an N × D matrix. The mathematical model of the population is given by formula (1):

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times M} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,M} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,M} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,M} \end{bmatrix}_{N\times M} \tag{1}$$

Here, the population N and individual's position are represented by X and X_i, respectively, $X_{i,j}$ denotes the value on the j_{th} dimension and M are the number of individuals and the largest dimension. The objective function is represented by the fitness function.

2.2 Prey Identification and Attack

The search and attack of prey is the first stage of the NGO in each iteration, which simulates the optimal solution as prey and each individual launches the attack it. In this stage, every individual identifies the location of the current best solution and updates their position based on this optimal solution. The main purpose is to enable the northern goshawk individual to search the feasible solution space more widely. Thus, global search is carried out. Formulas (2)–(4) represent the mathematical model for the initial phase.

$$P_i = X_Z \tag{2}$$

$$x_{i,j}^{new1} = \begin{cases} x_{i,j} + r\left(p_{i,j} - I * x_{i,j}\right), F_{p_i} < F_i \\ x_{i,j} + r\left(x_{i,j} - p_{i,j}\right), F_{p_i} \geq F_i \end{cases} \tag{3}$$

$$X_i = \begin{cases} x_i^{new1}, F_i^{new1} < F_i \\ X_i, F_i^{new1} \geq F_i \end{cases} \tag{4}$$

where $i = 1, 2, \cdots, N$, the value of I can be 1 or 2 and $r = \text{rand}(0, 1)$.. P_i is a candidate solution. F_{P_i} is its objective function value. x_i^{new1} is the new position of the first stage, $x_{i,j}^{new1}$ is its j_{th} dimension. F_i^{new1} represents objective function value of the first stage.

2.3 Chase and Escape Operation

The second stage is the chase and escape of prey, and the local exploitation is carried out by simulating the local activities of prey to affect the individual position of the northern goshawk. The position updating formula of the second stage is shown in formulas (5)–(6):

$$x_{i,j}^{new2} = x_{i,j} + R\left(2r - 1\right) x_{i,j} \tag{5}$$

$$X_i = \begin{cases} x_i^{new2}, F_i^{new2} < F_i \\ X_i, F_i^{new2} \geq F_i \end{cases} Z = 1, 2, \cdots I, \cdots N \tag{6}$$

where x_i^{new2} is the new position of the second stage and $x_{i,j}^{new2}$ is its value in j_{th} dimension. F_i^{new2} represents objective function value of the second stage. R is the hunting radius. The expression of R is shown in formula (7):

$$R = 0.02\left(1 - \frac{t}{T}\right) \tag{7}$$

where t denotes the iteration number, and T signifies the maximum iteration number. R will diminish as the iteration count increases, indicating that the hunting radius of the NGO contracts in the later stages of iteration, thereby enhancing the algorithm's local exploitation capability. This mechanism is pivotal to NGO achieving superior optimization accuracy compared to many other.

Each iteration within the NGO process encompasses these two phases. The prevailing optimal solution serves as a beacon for other members to gravitate towards improved outcomes. Upon reaching the maximum iteration limit, the algorithm concludes. Fig. 1 illustrates the flowchart of the standard NGO.

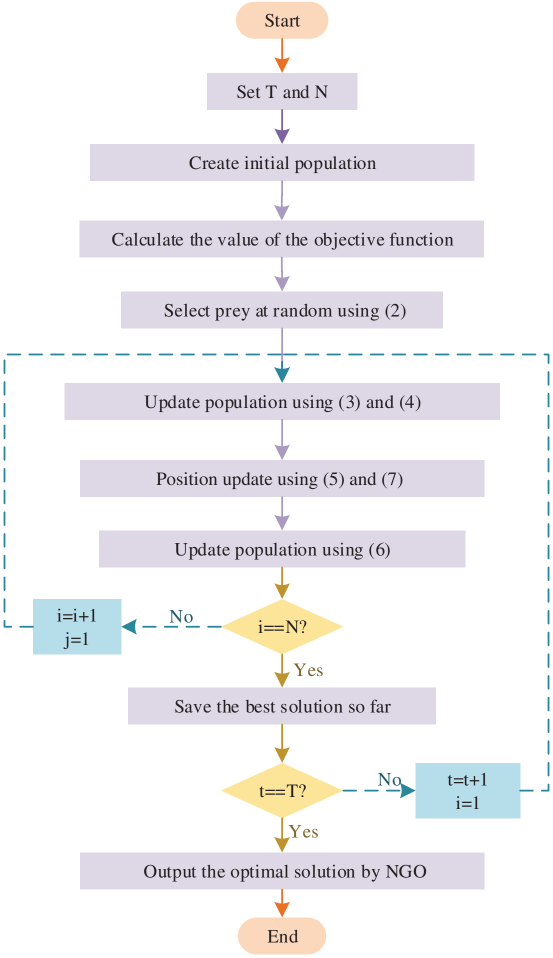
Start
Set T and N
Create initial population
Calculate the value of the objective function
Select prey at random using (2)
Update population using (3) and (4)
Position update using (5) and (7)
Update population using (6)
i==N?
No
i=i+1
j=1
Yes
Save the best solution so far
t==T?
No
t=t+1
i=1
Yes
Output the optimal solution by NGO
End

Figure 1: The flowchart of the northern Goshawk algorithm

3 The Proposed Algorithm: INGO

The essence of swarm intelligence algorithm lies in the algorithm's capability to explore and exploit. The improvement of the so-called swarm intelligence algorithm is to help algorithms to strike a balance between the exploration and exploitation. This facilitates the algorithm's capacity to not only broadly search the feasible solution space but also to refine its approach in achieving a closer approximation to the optimal solution with greater accuracy. Based on previous research, this section presents four effective improvement strategies for function optimization of NGO. By combining all four strategies, the NGO is strengthened in terms of its ability to explore and exploit.

3.1 Tent Chaotic Mapping

Given the lack of prior knowledge about the global optimal solution for the optimization problem, it is advisable to distribute the population evenly throughout the search space. However, the randomly generated initial population is frequently distributed unevenly, thereby impacting the diversity of the population. Although the intelligent algorithm is insensitive to the initialization method and there are no notable disparities among the methods of population initialization after enough iterations [40], using a more uniform initialization technique will improve the algorithm's exploration ability during the early iteration.

Chaotic sequence is used to generate initial population instead of randomly generated initial population in this paper. Tent mapping is widely used in the initial improvement of swarm intelligent algorithm due to its ability to generate more uniform chaotic sequence [41]. The iterative formula of tent mapping operator adopted is shown as formula (8):

$$X_{t+1} = \begin{cases} 2X_t, X_t \in [0, 0.5] \\ 2\left(1 - X_t\right), X_t \in (0.5, 1] \end{cases} \tag{8}$$

3.2 Whale Fall

The convergence speed of NGO is quick, which it also makes the local optimal value affecting the algorithm tremendously in the early stage, resulting in the low global search ability of it. In the first stage of NGO search, there is a phenomenon of rapid convergence, and it is quite probable to fall into local optimality. Therefore, improving the global exploratory ability is of great significance to improve NGO performance.

Targeted addition of some strategies is often an effective technique to improve the global exploratory capability of the algorithm. Whale fall refers to the phenomenon wherein a whale dies and its body descends to the bottom of the ocean and it is one of the main reasons life in the deep ocean flourishes. The whale fall phenomenon was introduced into the beluga optimization algorithm. The BWO gives a model of the falling behavior of the whale after death in each iteration. An updated position is established by using the beluga whale's position and the length of its fall, along with a tiny probability.

Taking into account NGO's inclination to get stuck in local optimum, this paper incorporated the whale fall phenomenon into the first stage search of it, representing the behavior of the northern goshawk, which has a very small probability of falling or was affected by other random individuals for other reasons after the large-scale exploration behind the first stage search, assists the algorithm in increasing its chances of breaking free from local optimal values. Formula (9) expresses the mathematical model of the whale fall.

$$X_i\left(t + 1\right) = r_1 X_i\left(t\right) - r_2 X_R\left(t\right) + r_3 X_{step} \tag{9}$$

where $r_1, r_2, r_3 \in (0, 1)$ are random numbers, X_R is random individuals, and the mathematical expression of X_{step} is illustrated in formula (10):

$$X_{step} = \left(ub - lb\right) e^{(-ct/T)} \tag{10}$$

where ub is the upper bounds and lb is the upper bounds of variables. C is related to whale landing probability W_f and population size N. The mathematical expression of C and W_f is given by formula (11):

$$\begin{cases} C = 2W_f N \\ W_f = 0.1 - \dfrac{t}{20T} \end{cases} \tag{11}$$

In the process of each iteration of the NGO, the whale landing operation is carried out after the completion of the prey discovery and attack stage, which can better conduct a global search, so that the algorithm approaches the global optimal solution with a greater probability.

3.3 T-Distribution Variation

Variation is the most common strategy used for swarm intelligence algorithms to fall into local optimality. Cauchy variation and Gaussian variation are common. According to the characteristics of the NGO, this paper adopts T-distribution mutation strategy [42] to improve its weaknesses. The algorithm's iteration count serves as the degree of freedom for the T-distribution. Following each individual's completion of the initial two search stages, a variation operation is performed on the optimal solution. The fitness value of the optimal solution before and after variation is compared to re-evaluate the optimal solution and retain it. The variation formula is shown in formula (12):

$$X_i\,(new) = (T + 1)\,X_i \tag{12}$$

Here, T is a T-distribution. The mathematical formula for the T-distribution is shown in formula (13):

$$f_z\,(x) = \frac{\Gamma\left(\frac{t+1}{2}\right)}{\sqrt{n\pi}\,\Gamma\left(\frac{t}{2}\right)}\left(1 + \frac{x^2}{t}\right)^{-\frac{t+1}{2}} \tag{13}$$

Here, t is the number of iterations.

During the initial phase of iteration, the T-distribution resembles the Cauchy distribution, mitigating the impact of local extremities. This alteration increases the likelihood of the algorithm breaking free from local optima, thereby boosting its global exploratory capabilities. In the subsequent phases, the T-distribution approaches the characteristics of a Gaussian distribution, facilitating improved local exploitation and enhanced convergence precision for the algorithm, while simultaneously quickening the rate of convergence.

3.4 Added the State Transition Algorithm (STA)

Algorithm fusion is the trend of swarm intelligence algorithm improvement recently. The main purpose is to combine the advantages of various algorithms so that new algorithms have stronger exploration and exploitation ability. STA is a competitive intelligent algorithm proposed by Zhou in 2012. It uses the state space expression as the framework to design four transformation operators, considering the process of the feasible solution iteration as a process of state transition. Based on the current optimal solution, some state transition operator is run, and the candidate solution set is generated by generating neighborhood and sampling in the neighborhood, and then compared with the current best solution, a new optimal solution is obtained.

STA utilizes the state space expression as a framework to simulate the iterative updating process of the solution, as shown in formula (14):

$$\begin{aligned} x_{t+1} &= A_t x_t + B_t u_t \\ y_{t+1} &= f\,(x_{t+1}) \end{aligned} \tag{14}$$

Here, $x_t = [x_1, x_2, \cdots x_i, \cdots, x_n]^T$ represents a candidate solution, A_t and B_t are state transition matrices, which can be random matrices, u_t is the state function. $f\,(x_t)$ is the objective function.

The expression of the rotation transformation operator is shown in formula (15):

$$x_{t+1} = x_t + \alpha \frac{1}{n \left\| x_t \right\|_2} R_r x_t \tag{15}$$

Here, $R_r \in \mathrm{R}^{n\times n}$ is the random matrix uniformly distributed between $[-1, 1]$. $\|\cdot\|_2$ is the norm. $\alpha > 0$ is the rotation factor.

The expression of translation transformation operator is shown in formula (16):

$$x_{t+1} = x_t + \beta R_t \frac{x_t - x_{t-1}}{\left\| x_t - x_{t-1} \right\|_2} \tag{16}$$

Here, $R_r \in \mathrm{R}$ is the random number uniformly distributed between $[0, 1]$. $\beta > 0$ is the translation factor.

The expression of expansion transformation operator is shown in formula (17):

$$x_{t+1} = x_t + \gamma R_e x_t \tag{17}$$

Here, $R_e \in \mathrm{R}^{n\times n}$ is the value of non-zero element following the random diagonal matrix of Gaussian distribution. $\gamma > 0$ is the expanding factor.

The expression of axesion transformation operator is shown in formula (18):

$$x_{t+1} = x_t + \delta R_a x_t \tag{18}$$

Here, $R_a \in \mathrm{R}^{n\times n}$ is the value of non-zero element, which follows the sparse random diagonal matrix of Gaussian distribution. $\delta > 0$ is the axial factor. The axesion transformation operator improves the ability to search in a certain dimension.

For the existing optimal solution, a neighborhood is constructed through a candidate solution set generated by a specific transformation operator. Sampling within this neighborhood is executed via a defined sampling method, from which the superior solution is identified. Subsequently, optimal solutions undergo updates through greedy mechanisms; namely, comparing the derived solutions with the current optimal and selecting the most advantageous one. The greedy mechanism is shown in formula (19):

$$x_t = \begin{cases} x_t^{new}, if\ f\left(x_t^{new}\right) < f\left(x_t\right) \\ x_t, other \end{cases} \tag{19}$$

Aimed at the characteristics of NGO, which has high search speed but low search accuracy, STA is introduced into NGO to enhance its convergence accuracy, so that it could get better optimization results when solving function optimization problems. STA possesses strong local exploitation capability and is capable of thoroughly exploring the surrounding space of the current optimal solution. Upon discovering the best outcome during each NGO iteration, STA employs its four transformation operators to explore the vicinity of the optimal solution, thereby guaranteeing enhanced local exploitation of the algorithm.

3.5 The Pseudo-Code and Flowchart of INGO

Algorithm 1 outlines INGO's pseudo-code. At line 3, the Tent chaotic map initializes the population. Line 11 updates positions through the whale fall strategy, and line 18 applies the T-distribution for variations. STA's utilization for neighborhood searches of the optimal solution is marked at line 20.

Algorithm 1: Pseudo-code of INGO algorithm

```
Start INGO
  Enter the information for the objective function.
  Define the values of parameters $T$ and $N$
  The initial position of INGO is generated using Tent chaotic mapping
  for 1 $t = 1 : T$
    for 2 $i = 1 : N$
      Select the prey using (2).
      for 3 $j = 1 : m$
        Update the position of each dimension using (3).
      end 3
      Update the position of the optimal solution using (4).
      Whale fall strategy: Use formulas (9)–(11) for location updates
      Calculate the value of $R$using (5)
      for 4 $j = 1 : m$
        Update the position of each dimension using (7).
      end 4
      Update the position of the optimal solution using (6).
      for 5 $j = 1 : m$
        T-distribution variation strategy: Use formulas (12) and (13) for location updates
      end 5
      STA strategy: Use STA for local search
    end 2
    Save the optimal solution.
  end 1
  Output best solution.
end INGO.
```

Fig. 2 displays the flowchart of INGO. Upon setting the algorithm's maximum iterations T and population size N, execution commences. Initially, population generation via Tent chaotic mapping, individual fitness evaluation, and sorting to identify the optimum position sets the stage. Subsequent phases leverage prey identification and the attack strategy for positional updates and the whale-fall technique to enhance escape from local optima. The NGO's chase and escape mechanism then refines accuracy, followed by a T-distribution variation to intensively explore the solution space. Conclusively, sampling near the optimum with the state transition algorithm identifies a superior solution, ensuring progression to the next iteration once neighborhood superiority is confirmed. The algorithm concludes once it fulfills the stopping criteria.

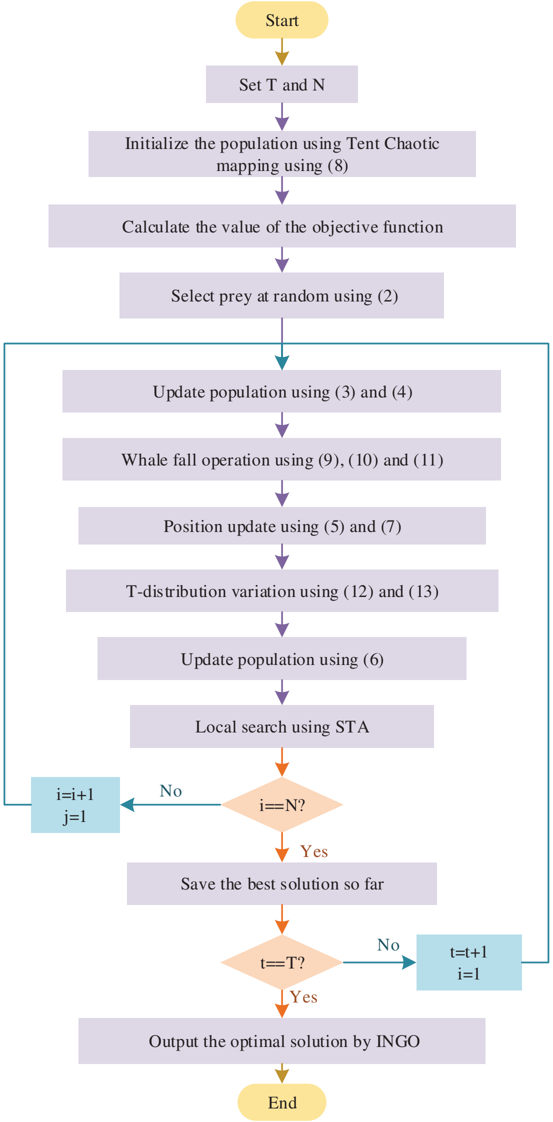
Start
Set T and N
Initialize the population using Tent Chaotic mapping using (8)
Calculate the value of the objective function
Select prey at random using (2)
Update population using (3) and (4)
Whale fall operation using (9), (10) and (11)
Position update using (5) and (7)
T-distribution variation using (12) and (13)
Update population using (6)
Local search using STA
i==N?
No
i=i+1
j=1
Yes
Save the best solution so far
t==T?
No
t=t+1
i=1
Yes
Output the optimal solution by INGO
End

Figure 2: Flowchart of the INGO

4 Experimental Results

4.1 Experimental Setup

This section mainly analyzes the performance of INGO. Firstly, the effectiveness of the strategy is analyzed by gradually adding the strategy, secondly, the scalability of the algorithm is verified by testing on different dimensions, and finally, the algorithm is compared with other algorithms on 23 benchmark functions [43] and CEC2017 test set [44]. The algorithms involved in the comparison of these 53 benchmark functions are: AO, Hunter–prey Optimization (HPO) [45], Sand Cat Swarm Optimization (SCSO) [46], HHO, WOA, GWO, SCA, DA, Salp Swarm Algorithm (SSA) [47], BWO and an Improved Northern Goshawk Optimization (INGOfu) [48]. These algorithms can be categorized into four types: Classical swarm intelligence algorithms that are widely applied (GWO, WOA, DA, SSA), which have garnered significant attention from researchers; physics-based algorithms (SCA), with SCA often being utilized in conjunction with other swarm intelligence algorithms; recently developed competitive swarm intelligence algorithms (AO, HPO, SCSO, HHO), which, when compared with newer algorithms, can better highlight the performance of INGO; and the improved NGO algorithm (INGOfu). The diversity of the types of algorithms involved in the comparison provides a better and more comprehensive assessment of INGO's performance. The population of all algorithms is 50, with a maximum iteration count of 1000 and a dimension of 30. The information and parameters of the algorithms used in this paper are given in Table 1. Run the algorithm independently for 30 times and compare the mean value and standard deviation. All algorithm tests were carried out on Windows 10(64-bit), Intel(R) Core(TM) i7-11800H, CPU@2.30 GHz, RAM:16 G MATLAB 2021b.

Table 1: The information and parameters of the algorithms used in this paper

Algorithm	Parameter
AO	α, β: 0.1
HPO	C: [1,0.02], β: 0.1
SCSO	Sensitivity range: [2,0]
HHO	Escaping probability: 0.5 Escaping energy: 0.5
WOA	a: [2, 0], r: rand (0, 1), l: rand (-1, 1)
GWO	a: [2, 0], A: [2, 0], C: 2*rand (0, 1)
SCA	N/A
DA	r_1, r_2: rand (0, 1), s, $a = 0.1$, $c = 0.7$ f, $e = 1$, $w \in (0.2,0.9)$
SSA	p: 0.5
BWO	N/A
INGOfu	$n = 1.02e + 04$

4.2 The Impact of Optimization Strategies on Algorithm

A set of experiments is set up in this section to analyze the effect of the four strategies introduced in INGO algorithm on the algorithm. Four strategies are added to NGO successively, and different combinations are shown in Table 2, where A represents tent chaotic mapping strategy, B represents whale fall strategy, C represents T-distribution variation strategy, D represents neighborhood search strategy, 1 represents these strategies are added to the NGO algorithm, and four new algorithms are obtained: NGO-A, NGO-B, NGO-C and NGO-D. NGO-D is equivalent to our proposed INGO algorithm. At the same time, NGO and these four algorithms were compared and tested on part of the benchmark functions. Table 3 shows the average and standard deviation test results of the 12 functions F1–F6 and F8–F13 out of 23 benchmark functions of 5 different algorithms. It is worth mentioning that, all the five algorithms F14–F23 have obtained the optimal values. Table 4 displays the performance outcomes of five algorithms across 30 functions within the CEC2017 test suite, with the reported results reflecting the mean of 30 independent trials. The data reveal a progressive improvement in algorithmic performance correlating with the incremental integration of strategies, thereby validating the efficacy of the four strategies. This enhancement signifies their collective capability to augment the algorithm's search efficiency.

Table 2: Combination of strategies

Name	A	B	C	D
NGO	0	0	0	0
NGO-A	1	0	0	0
NGO-B	1	1	0	0
NGO-C	1	1	1	0
NGO-D	1	1	1	1

Table 3: The results of the five algorithms on benchmark functions

Index	FUN	F1	F2	F3	F4	F5	F6
NGO	MEAN	1.27E−88	7.12E−46	1.87E−20	1.26E−37	2.53E+01	1.94E−05
	STD	1.86E−88	4.71E−46	1.02E−19	1.56E−37	4.20E−01	1.02E−04
NGO-A	MEAN	1.33E−88	9.78E−46	5.52E−22	1.25E−37	2.52E+01	1.41E−05
	STD	2.08E−88	7.57E−46	1.54E−21	1.17E−37	4.05E−01	5.76E−06
NGO-B	MEAN	5.82E−89	2.84E−46	8.74E−23	3.25E−38	1.38E−06	3.09E−06
	STD	1.49E−88	2.78E−46	4.25E−22	2.05E−38	4.52E−06	2.14E−06
NGO-C	MEAN	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.23E−07	7.13E−07
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	5.09E−07	3.75E−07
NGO-D	MEAN	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**3.87E−07**	**2.53E−10**
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**7.23E−07**	**6.41E−11**
Index	FUN	F8	F9	F10	F11	F12	F13
NGO	MEAN	−7.69E+03	**0.00E+00**	5.98E−15	**0.00E+00**	1.04E−06	6.36E−02
	STD	4.87E+02	**0.00E+00**	1.79E−15	**0.00E+00**	4.41E−06	6.82E−02
NGO-A	MEAN	−7.64E+03	**0.00E+00**	7.21E+00	**0.00E+00**	8.82E−07	7.01E−02
	STD	1.36E+03	**0.00E+00**	7.56E+00	**0.00E+00**	4.44E−07	5.86E−02
NGO-B	MEAN	−3.17E+41	**0.00E+00**	6.10E−15	**0.00E+00**	7.63E−08	1.81E−10
	STD	1.58E+42	**0.00E+00**	1.80E−15	**0.00E+00**	1.35E−07	2.76E−10
NGO-C	MEAN	−**6.23E+23**	**0.00E+00**	8.88E−16	**0.00E+00**	2.11E−08	1.52E−10
	STD	**2.66E+24**	**0.00E+00**	1.00E−31	**0.00E+00**	1.29E−08	2.23E−10
NGO-D	MEAN	−1.70E+23	**0.00E+00**	**8.88E−16**	**0.00E+00**	**2.88E−11**	**7.10E−11**
	STD	7.68E+23	**0.00E+00**	**1.00E−31**	**0.00E+00**	**2.02E−11**	**4.52E−11**

Table 4: The average of the test results of the 5 algorithms on CEC2017 benchmark functions

Name	NGO	NGO-A	NGO-B	NGO-C	NGO-D
F1	2.636E+06	4.271E+06	1.621E+05	1.319E+05	**4.026E+03**
F2	5.091E+16	5.719E+18	1.003E+16	1.466E+16	**1.014E+08**
F3	6.222E+04	6.157E+04	4.998E+04	4.104E+04	**1.926E+04**
F4	5.184E+02	5.117E+02	5.124E+02	4.973E+02	**4.844E+02**
F5	6.836E+02	6.676E+02	6.770E+02	6.362E+02	**6.153E+02**
F6	6.256E+02	6.266E+02	6.058E+02	6.080E+02	**6.017E+02**
F7	9.855E+02	9.833E+02	9.377E+02	8.950E+02	**8.792E+02**
F8	9.528E+02	9.463E+02	9.420E+02	9.351E+02	**9.172E+02**
F9	3.379E+03	3.464E+03	1.647E+03	1.621E+03	**1.301E+03**
F10	5.431E+03	5.647E+03	6.135E+03	6.220E+03	**4.443E+03**
F11	1.235E+03	1.246E+03	1.243E+03	1.234E+03	**1.217E+03**
F12	1.201E+06	1.336E+06	1.936E+06	1.654E+06	**1.100E+06**
F13	1.951E+04	2.350E+04	7.213E+03	**6.283E+03**	8.851E+03
F14	1.276E+04	**1.272E+04**	2.550E+04	1.696E+04	1.577E+04
F15	4.986E+03	4.726E+03	3.147E+03	3.773E+03	**3.009E+03**
F16	2.750E+03	2.724E+03	3.138E+03	2.628E+03	**2.489E+03**
F17	2.038E+03	1.999E+03	2.024E+03	1.946E+03	**1.915E+03**
F18	1.532E+05	1.834E+05	7.345E+05	3.544E+05	**2.357E+05**
F19	5.196E+03	6.539E+03	5.851E+03	5.953E+03	**5.146E+03**
F20	2.439E+03	2.412E+03	2.589E+03	2.440E+03	**2.354E+03**
F21	2.441E+03	2.429E+03	2.387E+03	2.381E+03	**2.368E+03**
F22	2.313E+03	2.653E+03	2.317E+03	**2.305E+03**	**2.300E+03**
F23	2.797E+03	2.797E+03	2.752E+03	2.750E+03	**2.730E+03**
F24	2.933E+03	2.935E+03	2.948E+03	**2.916E+03**	**2.919E+03**
F25	2.932E+03	2.928E+03	2.897E+03	2.896E+03	**2.891E+03**
F26	3.899E+03	5.179E+03	**3.894E+03**	4.256E+03	4.349E+03
F27	3.227E+03	3.230E+03	**3.200E+03**	**3.200E+03**	**3.200E+03**
F28	**3.303E+03**	**3.299E+03**	**3.300E+03**	**3.300E+03**	**3.300E+03**
F29	4.040E+03	3.906E+03	3.833E+03	3.730E+03	**3.666E+03**
F30	2.205E+04	2.276E+04	1.856E+04	1.690E+04	**1.216E+04**

4.3 The Impact of Problem Dimensions

The dimensionality of optimization problems exerts a discernible influence on the operational efficiency of algorithms. The intricacy of high-dimensional optimization problems serves as a more effective measure for assessing an algorithm's performance. In this section, the first 13 functions of 23 benchmark functions (F1–F13) are used to conduct dimensional tests on INGO. The performance of NGO and INGO was tested on four dimensions: 30, 50, 100 and 500. Table 5 shows the test results.

The results show that the performance of both algorithms deteriorates with the increase of dimension, while INGO is less affected and can still find better solutions except for function F7, which proves that after the improvement of four strategies, INGO still has a strong development and exploration ability even when dealing with high-dimensional problems.

Table 5: The results of NGO and INGO in different dimensions on 13 classical benchmark functions

F/dim	30		50		100		500	
	NGO	INGO	NGO	INGO	NGO	INGO	NGO	INGO
F1	2.35E−88	**0.00E+00**	1.50E−88	**0.00E+00**	2.11E−88	**0.00E+00**	2.33E−88	**0.00E+00**
F2	6.56E−46	**0.00E+00**	8.13E−46	**0.00E+00**	8.37E−46	**0.00E+00**	8.70E−46	**0.00E+00**
F3	1.92E−21	**0.00E+00**	3.15E−22	**0.00E+00**	4.51E−22	**0.00E+00**	5.05E−22	**0.00E+00**
F4	7.99E−38	**0.00E+00**	1.27E−37	**0.00E+00**	9.21E−38	**0.00E+00**	1.02E−37	**0.00E+00**
F5	2.55E+01	**2.40E−07**	2.54E+01	**1.65E−07**	2.54E+01	**2.30E−07**	2.54E+01	**2.57E−07**
F6	9.66E−07	**2.72E−10**	3.27E−06	**2.60E−10**	1.63E−06	**2.49E−10**	9.13E−07	**2.54E−10**
F7	**4.78E−04**	1.46E−02	**4.89E−04**	1.42E−02	**4.70E−04**	1.77E−02	**4.67E−04**	1.70E−02
F8	−7.75E+03	**−3.01E+24**	−7.74E+03	**−2.47E+25**	−7.67E+03	**−1.20E+39**	−7.69E+03	**−4.35E+24**
F9	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**
F10	5.74E−15	**8.88E−16**	5.39E−15	**8.88E−16**	5.63E−15	**8.88E−16**	6.10E−15	**8.88E−16**
F11	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**
F12	2.10E−7	**2.98E−11**	1.88E−07	**2.87E−11**	2.29E−07	**2.86E−11**	4.18E−07	**2.34E−11**
F13	7.55E−02	**5.54E−11**	7.75E−02	**7.83E−11**	7.75E−02	**7.79E−11**	6.07E−02	**6.42E−11**

4.4 Comparison with Other Algorithms on 23 Classical Benchmark Functions

INGO's function optimization performance is tested using 23 classic benchmark functions. There are 7 single-peak functions (F1–F7), 6 multi-peak functions (F8–F13) and 10 fixed-dimension multi-peak test functions (F14–F23). The algorithm's exploitation ability is verified using unimodal functions. Multimodal function is used to reveal the algorithm's exploration ability. INGO is compared with the NGO, AO, HPO, SCSO, HHO, WOA, GWO, SCA and DA.

Table 6 presents the test results for functions F1–F7, encompassing both mean and standard deviation values. In a comparative analysis involving the initial seven benchmark functions against nine alternative algorithms, INGO consistently achieved the theoretical optimal value of zero across the first four functions and secured the leading position for functions F5 and F6, culminating in a composite ranking of first place. This underscores INGO's superior exploratory capabilities over the existing NGO algorithm, alongside enhanced local convergence precision, attributed to the refined integration of the STA strategy that bolsters INGO's local exploratory proficiency. Notably, INGO's performance was subpar on function F7, a function whose complexity escalates linearly with dimensionality and emulates noise encountered in real-world scenarios. This underperformance on such problems indicates a deficiency in INGO's learning capacity during the optimization process, marked by an inability to make dynamic adjustments during the search. Future enhancements of the algorithm could consider the incorporation of dynamic parameter adjustments to facilitate adaptive modifications, thereby augmenting its optimization efficacy.

Table 6: Fitness results of different algorithms on unimodal functions (F1–F7)

Function	Index	INGO	NGO	AO	HPO	SCSO	HHO	WOA	GWO	DA	SCA
F1	MEAN	**0.00E+00**	1.02E−177	5.18E−200	0.00E+00	1.10E−231	1.01E−183	5.36E−147	6.18E−59	1.29E+03	3.21E−02
	STD	**0.00E+00**	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.92E−146	2.16E−58	6.18E+02	6.63E−02
F2	MEAN	**0.00E+00**	1.37E−92	6.60E−100	2.61E−186	4.33E−123	5.46E−95	2.02E−101	1.09E−34	1.55E+01	2.03E−05
	STD	**0.00E+00**	2.11E−92	3.28E−99	0.00E+00	1.35E−122	2.98E−94	1.10E−100	1.60E−34	5.87E+00	4.60E−05
F3	MEAN	**0.00E+00**	1.65E−49	7.13E−199	1.14E−299	1.15E−197	4.38E−144	1.96E+04	1.07E−13	9.77E+03	4.49E+03
	STD	**0.00E+00**	5.47E−49	0.00E+00	0.00E+00	0.00E+00	2.40E−143	1.14E+04	3.27E−13	5.57E+03	4.56E+03
F4	MEAN	**0.00E+00**	2.38E−76	1.30E−103	1.74E−153	1.32E−103	6.03E−94	4.93E+01	1.51E−14	2.04E+01	2.09E+01
	STD	**0.00E+00**	4.66E−76	6.92E−103	9.51E−153	4.33E−103	2.42E−93	3.09E+01	1.52E−14	7.45E+00	1.19E+01
F5	MEAN	**3.00E−06**	2.37E+01	1.08E−03	2.17E+01	2.77E+01	2.62E−03	2.71E+01	2.69E+01	1.54E+05	9.46E+02
	STD	**5.14E−06**	5.25E−01	1.24E−03	5.03E−01	8.33E−01	3.69E−03	3.84E−01	7.52E−01	1.57E+05	2.21E+03
F6	MEAN	**2.46E−11**	6.59E−08	1.74E−05	4.00E−11	1.96E+00	4.44E−05	4.25E−02	5.54E−01	1.35E+03	4.68E+00
	STD	**4.20E−11**	6.37E−08	4.72E−05	2.11E−10	5.77E−01	5.82E−05	5.83E−02	3.35E−01	7.14E+02	3.74E−01
F7	MEAN	3.28E−02	2.77E−04	**4.13E−05**	9.67E−05	1.00E−04	7.76E−05	2.08E−03	8.55E−04	3.47E−01	3.30E−02
	STD	2.69E−02	9.17E−05	**4.19E−05**	9.83E−05	1.11E−04	6.09E−05	2.24E−03	6.16E−04	1.73E−01	1.96E−02

The test results for the multimodal functions F8–F13 are displayed in Table 7. Across these six multimodal functions, INGO consistently ranks first in comprehensive comparisons with other algorithms, indicating its superior exploratory capabilities and broader search scope within the feasible solution space. This is attributed to the enhanced global search ability of the algorithm, which is a result of the T-distribution mutation and the improved whale landing behavior strategies. Notably, INGO's performance on F12 was slightly inferior to that of HPO, ranking second.

Table 7: Fitness results of different algorithms on multimodal functions (F8–F13)

Function	Index	INGO	NGO	AO	HPO	SCSO	HHO	WOA	GWO	DA	SCA
F8	MEAN	**−4.36E+30**	−7.93E+03	−1.01E+04	−9.29E+03	−6.73E+03	−1.26E+04	−1.06E+04	−5.99E+03	−5.81E+03	3.87E+03
	STD	**2.39E+31**	5.26E+02	3.62E+03	6.69E+02	6.97E+02	2.84E−01	1.88E+03	8.00E+02	8.19E+02	2.55E+02
F9	MEAN	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.79E−15	4.50E−01	1.67E+02	1.56E+01
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.44E−14	1.51E+00	3.37E+01	2.41E+01
F10	MEAN	**8.88E−16**	5.86E−15	**8.88E−16**	**8.88E−16**	**8.88E−16**	**8.88E−16**	4.09E−15	1.69E−14	8.99E+00	9.90E+00
	STD	**1.00E−31**	1.77E−15	**1.00E−31**	**1.00E−31**	**1.00E−31**	**1.00E−31**	1.71E−15	2.91E−15	1.47E+00	9.49E+00
F11	MEAN	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.86E−03	2.34E−03	1.13E+01	2.91E−01
	STD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.02E−02	6.46E−03	4.84E+00	2.70E−01
F12	MEAN	1.56E−12	5.33E−09	1.11E−06	**3.82E−13**	1.02E−01	1.44E−06	7.71E−03	3.91E−02	1.60E+02	4.05E+01
	STD	1.19E−12	4.51E−09	1.82E−06	**1.54E−12**	5.00E−02	1.57E−06	9.67E−03	2.04E−02	4.15E+02	1.99E+02
F13	MEAN	**1.38E−11**	7.88E−02	5.21E−06	9.54E−02	2.38E+00	1.69E−05	2.83E−01	4.79E−01	7.00E+04	6.34E+00
	STD	**1.41E−11**	8.99E−02	6.59E−06	1.36E−01	3.60E−01	2.09E−05	2.52E−01	2.05E−01	1.27E+05	8.19E+00

The experimental outcomes for functions F14–F23 are detailed in Table 8. The optimization results of INGO and NGO are infinitely close to the theoretical optimal values, demonstrating their superior optimization precision and stability over other algorithms. Across these ten functions, INGO consistently ranked first, with both NGO and INGO achieving the theoretical optimal values on seven of the test functions. This underscores the unique advantage of these two algorithms in addressing the optimization problems of fixed-dimensional multimodal functions. Although the original NGO algorithm matched INGO in achieving the optimal mean values across the ten functions, INGO exhibited a lower standard deviation, indicating greater reliability. This aspect is one of the key factors in evaluating algorithm performance.

Table 8: Fitness results of different algorithms on fixed dimensional multimodal functions (F14–F23)

Function	Index	INGO	NGO	AO	HPO	SCSO	HHO	WOA	GWO	SCA	DA
F14	MEAN	**9.98E−01**	**9.98E−01**	1.98E+00	2.50E+00	3.48E+00	1.03E+00	2.02E+00	4.45E+00	1.03E+00	1.26E+00
	STD	**3.39E−16**	**3.39E−16**	2.72E+00	2.93E+00	3.79E+00	1.81E−01	2.00E+00	4.71E+00	1.81E−01	6.86E−01
F15	MEAN	**3.07E−04**	**3.07E−04**	4.28E−04	3.06E−03	3.78E−04	3.61E−04	6.55E−04	3.71E−03	4.38E−03	1.01E−03
	STD	**3.93E−19**	6.87E−08	6.11E−05	6.91E−03	2.48E−04	1.84E−04	3.56E−04	7.58E−03	7.14E−03	3.76E−04
F16	MEAN	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**
	STD	**0.00E+00**	**0.00E+00**	1.84E−04	**0.00E+00**	6.60E−11	1.17E−11	3.92E−11	5.26E−09	4.54E−06	3.19E−05
F17	MEAN	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	3.99E−01
	STD	**1.13E−16**	**1.13E−16**	8.14E−05	**1.13E−16**	2.08E−08	7.31E−07	3.30E−07	8.21E−06	6.07E−06	2.05E−03
F18	MEAN	**3.00E+00**	**3.00E+00**	3.01E+00	5.70E+00	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**
	STD	**4.52E−16**	**4.52E−16**	1.48E−02	1.48E+01	2.92E−06	5.40E−08	2.64E−05	9.17E−06	3.25E−05	1.94E−05
F19	MEAN	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**
	STD	**2.71E−15**	**2.71E−15**	2.61E−03	2.00E−03	3.26E−03	2.82E−03	4.35E−03	2.32E−03	3.67E−05	2.09E−03
F20	MEAN	**−3.32E+00**	**−3.32E+00**	−3.19E+00	−3.26E+00	−3.18E+00	−3.14E+00	−3.27E+00	−3.25E+00	−3.26E+00	−2.88E+00
	STD	**1.36E−15**	**1.36E−15**	8.88E−02	8.33E−02	2.83E−01	8.15E−02	8.36E−02	7.84E−02	7.13E−02	3.89E−01
F21	MEAN	**−1.02E+01**	**−1.02E+01**	−1.01E+01	−7.21E+00	−6.08E+00	−5.05E+00	−7.85E+00	−9.31E+00	−7.53E+00	−2.48E+00
	STD	**1.81E−15**	7.64E−07	5.25E−03	2.92E+00	2.07E+00	1.00E−03	2.71E+00	1.92E+00	2.69E+00	2.46E+00
F22	MEAN	**−1.04E+01**	**−1.04E+01**	**−1.04E+01**	−7.12E+00	−6.33E+00	−5.43E+00	−8.71E+00	**−1.04E+01**	−7.19E+00	−3.57E+00
	STD	**0.00E+00**	4.68E−07	9.98E−03	2.74E+00	2.29E+00	1.29E+00	2.64E+00	2.78E−04	3.17E+00	2.04E+00
F23	MEAN	**−1.05E+01**	**−1.05E+01**	**−1.05E+01**	−7.21E+00	−7.69E+00	−5.47E+00	−8.43E+00	**−1.05E+01**	−7.74E+00	−4.04E+00
	STD	**9.03E−15**	**9.03E−15**	7.87E−03	2.78E+00	2.99E+00	1.31E+00	2.85E+00	2.78E−04	3.36E+00	1.65E+00

Convergence curves offer a more intuitive reflection of an algorithm's optimization process. The partial convergence curves for unimodal and multimodal functions are depicted in Figs. 3 and 4, respectively. These figures reveal that, among the first thirteen test functions, INGO consistently provides more accurate solutions than its algorithmic counterparts. Furthermore, INGO exhibits a faster convergence rate on the majority of these functions, indicative of the positive impact derived from the integration of Tent chaotic mapping. This inclusion significantly enhances the diversity of the initial population, thereby broadening and equalizing the search across the feasible solution space in the early phases of iteration. Such observations underscore the efficacy of this strategy in augmenting the NGO algorithm.

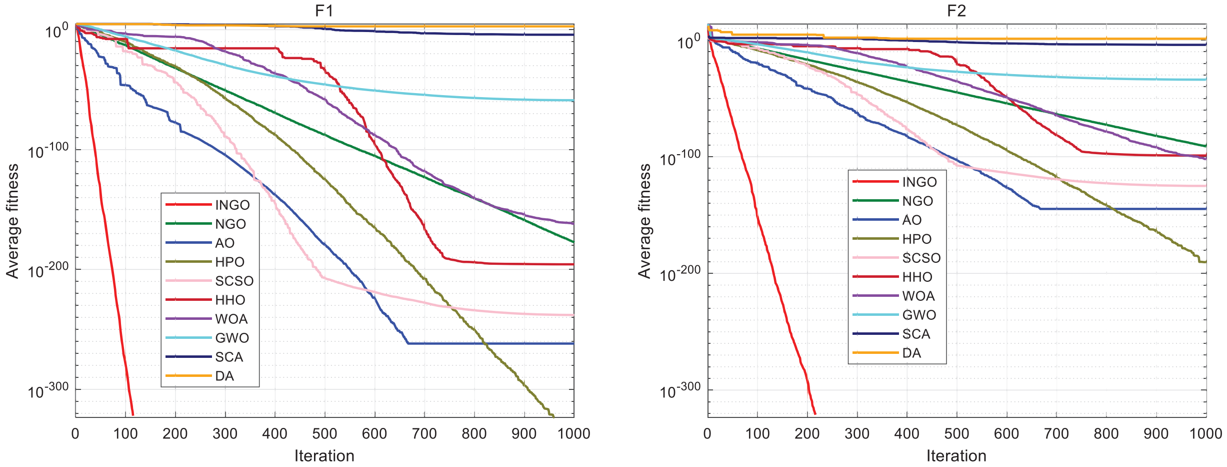

Figure 3: (Continued)

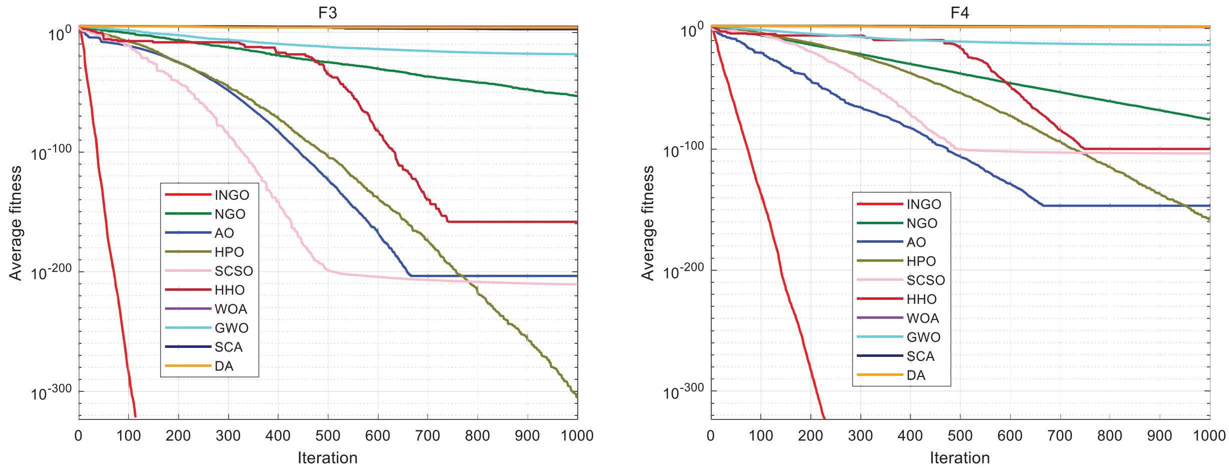

Figure 3: Iteration curve of unimodal function

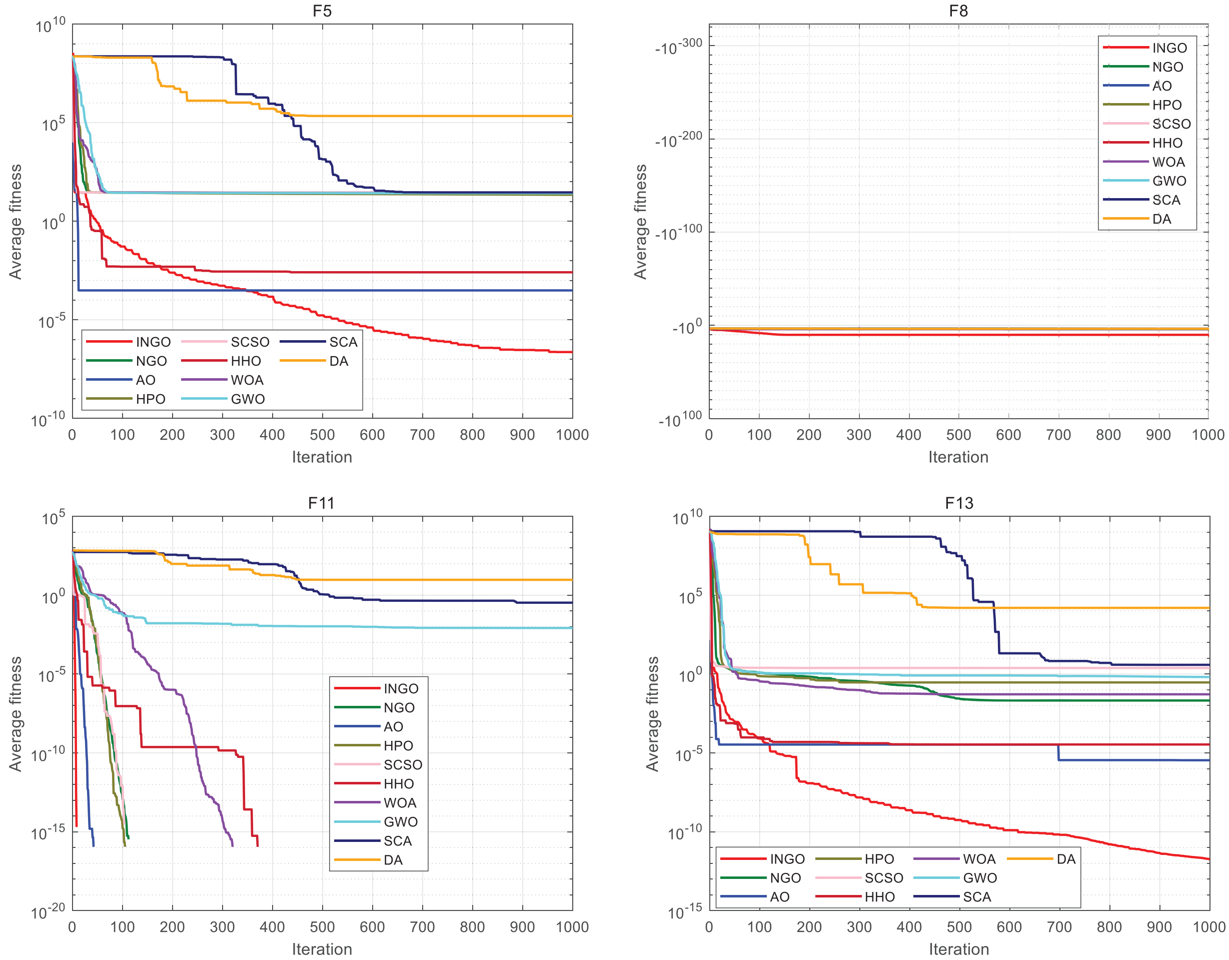

Figure 4: Iteration curve of multimodal function

To mitigate the impact of randomness on the evaluation of algorithm performance, we employed two statistical tests: The Wilcoxon signed-rank sum and the Friedman tests. Table 9 displays the statistical analysis results for 23 benchmark functions. Within the context of the Wilcoxon rank-sum test at a significance level of 0.05, data points with *p*-values exceeding 0.05 have been highlighted, with an aggregate statistical analysis conducted. The notation "=" denotes no significant difference in performance between INGO and the comparison algorithms on the given problem; "+" signifies that INGO outperformed the comparison algorithms; "−" indicates that the comparison algorithms surpassed INGO; "NaN" suggests that both algorithms produced identical outcomes across all 30 independent runs, evidencing no performance variation for the specified function problem. Out of a total of 207 results, INGO achieved 157 instances of "+", 7 of "−", 20 of "=", and 23 of "NaN". Results from the Friedman tests demonstrate that INGO possessed the lowest Average Ranks (AVR) value of 2.61, thereby securing the top rank amongst ten algorithms in a comprehensive comparison.

Table 9: Statistical analysis on 23 benchmark functions

INGO(VS)	INGO	NGO		AO		HPO		SCSO	
		P	S	P	S	P	S	P	S
F1	N/A	1.21E−12	+	1.21E−12	+	NaN		1.21E−12	+
F2	N/A	1.21E−12	+	1.21E−12	+	1.21E−12	+	1.21E−12	+
F3	N/A	1.21E−12	+	1.21E−12	+	4.57E−12	+	1.21E−12	+
F4	N/A	1.21E−12	+	1.21E−12	+	1.21E−12	+	1.21E−12	+
F5	N/A	2.96E−11	+	8.99E−11	+	3.02E−11	+	3.02E−11	+
F6	N/A	3.02E−11	+	3.02E−11	+	2.20E−07	+	3.02E−11	+
F7	N/A	3.00E−11	–	3.02E−11	–	3.02E−11	–	3.02E−11	–
F8	N/A	3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
F9	N/A	NaN		NaN		NaN		NaN	
F10	N/A	4.17E−13	+	NaN		NaN		NaN	
F11	N/A	NaN	=	NaN		NaN		NaN	
F12	N/A	3.02E−11	+	3.02E−11	+	3.80E−08	+	3.02E−11	+
F13	N/A	3.00E−11	+	3.02E−11	+	3.28E−03	+	3.02E−11	+
F14	N/A	NaN		1.04E−12	+	4.27E−13	+	6.87E−13	+
F15	N/A	**3.34E−01**	=	1.21E−12	+	3.20E−13	+	1.20E−12	+
F16	N/A	NaN		1.21E−12	+	1.69E−14	+	1.69E−14	+
F17	N/A	NaN		1.54E−04	+	1.69E−14	+	1.16E−12	+
F18	N/A	NaN		1.21E−12	+	**3.34E−01**	=	1.21E−12	+
F19	N/A	NaN		**6.41E−01**	=	5.86E−11	+	1.53E−04	+
F20	N/A	NaN		1.21E−12	+	**3.40E−01**	=	**6.41E−01**	=
F21	N/A	NaN		1.21E−12	+	5.64E−13	+	1.21E−12	+
F22	N/A	NaN		9.30E−04	+	**1.51E−01**	=	1.54E−04	+
F23	N/A	NaN		1.21E−12	+	**1.51E−01**	=	**1.00E+00**	=
+/−/=		10/1/2		18/1/1		14/1/4		17/1/2	
ARV	2.61	4.20		4.04		4.61		5.72	
RANK	1	3		2		4		6	

(Continued)

Table 9 (continued)

HHO		WOA		GWO		SCA		DA	
P	S	P	S	P	S	P	S	P	S
1.21E−12	+	1.21E−12	+	1.21E−12	+	1.21E−12	+	1.21E−12	+
1.21E−12	+	1.21E−12	+	1.21E−12	+	1.21E−12	+	1.21E−12	+
1.21E−12	+	1.21E−12	+	1.21E−12	+	1.21E−12	+	1.21E−12	+
1.21E−12	+	1.21E−12	+	1.21E−12	+	1.21E−12	+	1.21E−12	+
3.69E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
3.02E−11	–	4.20E−10	–	5.49E−11	–	4.08E−11	+	**6.84E−01**	=
3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
NaN		**1.61E−01**	=	6.56E−04	+	1.21E−12	+	1.21E−12	+
NaN		1.42E−10	+	3.12E−13	+	1.21E−12	+	1.21E−12	+
NaN		**3.34E−01**	=	4.19E−02	+	1.21E−12	+	1.21E−12	+
3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
4.16E−14	+	7.32E−13	+	6.53E−13	+	1.20E−13	+	1.21E−12	+
1.21E−12	+	1.21E−12	+	1.21E−12	+	1.21E−12	+	1.21E−12	+
1.69E−14	+	1.69E−14	+	1.15E−12	+	1.10E−12	+	1.21E−12	+
7.60E−13	+	1.06E−12	+	1.21E−12	+	1.06E−12	+	7.47E−10	+
1.25E−05	+	1.21E−12	+	1.21E−12	+	2.21E−06	+	1.21E−12	+
2.05E−05	+	**6.41E−01**	=	1.91E−07	+	1.21E−12	+	1.33E−08	+
1.21E−12	+	**6.41E−01**	=	**1.00E+00**	=	**1.57E−01**	=	1.21E−12	+
1.21E−12	+	1.21E−12	+	1.21E−12	+	1.21E−12	+	1.21E−12	+
1.21E−12	+	**1.57E−01**	=	1.21E−12	+	**5.89E−02**	=	1.21E−12	+
1.21E−12	+	**1.57E−01**	=	1.21E−12	+	**6.41E−01**	=	1.21E−12	+
+/−/=	19/1/0	16/1/6		21/1/1		20/0/3		22/0/1	
ARV	4.96	5.98		6.50		8.50		7.89	
RANK	5	7		8		10		9	

4.5 Comparison with Other Algorithms on CEC2017 Benchmark Functions

INGO's performance was tested and evaluated using the IEEE CEC2017 test set from the Conference on Evolutionary Computation 2017, which contains 30 test functions. CEC2017 has greater computational complexity than the common 23 reference functions and CEC2014. Moreover, the wide range of function types demonstrates that CEC2017 effectively assesses algorithm performance. In this test set, NGO, AO, HPO, SCSO, SSA, BWO, HHO, DA and INGOfu are compared. Table 10 records the results of 10 different algorithms in CEC2017. The result shows that INGO ranks the first overall on 30 test functions, while 28 functions of INGO rank the first among 10 algorithms. NGO's original algorithm is narrowly ahead of INGO in only one function F26. The convergence curves of 10 different algorithms on the CEC2017 test set is shown by Fig. 5. It can be seen that INGO algorithm with four improved strategies has better exploration and exploitation ability, and stronger stability and robustness than other algorithms. Additionally, the performance on functions F26 and F28 did not achieve optimal results. Function F26, a high-dimensional nonlinear minimization problem, along with F28, which exhibits rotational characteristics, posed significant challenges. The suboptimal performance on these two functions indicates certain limitations within INGO's search strategy, suggesting deficiencies in refining search operations and escaping local optima, or a lack of flexibility in adaptive adjustment of search strategies. This necessitates the enhancement of the algorithm's adaptive adjustment capabilities, for instance, through the implementation of adaptive dynamic adjustment mechanisms to dynamically balance exploration and exploitation. Such improvements will guide future modifications to INGO.

Table 10: Fitness results of different algorithms on CEC2017

Function	Index	AO	HPO	SCSO	SSA	BWO	HHO	DA	INGOfu	NGO	INGO
F1	MEAN	1.010E+08	1.133E+04	3.912E+09	6.503E+03	1.874E+07	4.769E+08	1.927E+09	3.599E+09	6.239E+03	**4.335E+03**
	STD	3.902E+07	8.564E+03	2.521E+09	6.308E+03	3.348E+06	3.167E+08	1.501E+09	1.448E+09	6.065E+03	**5.454E+03**
F2	MEAN	3.850E+30	9.984E+07	1.723E+33	2.037E+16	4.557E+22	4.501E+29	6.095E+30	2.347E+38	2.644E+13	**2.052E+02**
	STD	1.833E+31	3.147E+08	4.633E+33	8.361E+16	2.482E+23	1.290E+30	2.445E+31	9.507E+38	8.524E+13	**1.023E+01**
F3	MEAN	4.516E+04	3.839E+03	4.593E+04	1.702E+04	2.706E+04	2.425E+05	4.436E+04	1.038E+05	4.475E+04	**3.174E+03**
	STD	4.902E+03	3.125E+03	9.292E+03	1.013E+04	5.425E+03	6.869E+04	1.079E+04	2.970E+04	5.751E+03	**2.071E+03**
F4	MEAN	5.965E+02	4.847E+02	6.866E+02	5.028E+02	5.420E+02	6.954E+02	5.753E+02	9.552E+02	4.961E+02	**4.811E+02**
	STD	4.266E+01	1.986E+01	1.340E+02	2.340E+01	2.656E+01	9.835E+01	4.627E+01	2.973E+02	2.559E+01	**2.337E+01**
F5	MEAN	6.840E+02	6.610E+02	7.190E+02	6.544E+02	7.381E+02	8.170E+02	6.023E+02	8.455E+02	6.443E+02	**5.827E+02**
	STD	3.235E+01	3.217E+01	4.224E+01	5.569E+01	3.147E+01	4.912E+01	2.907E+01	6.304E+01	1.907E+01	**2.343E+01**
F6	MEAN	6.524E+02	6.327E+02	6.608E+02	6.466E+02	6.626E+02	6.730E+02	6.083E+02	6.828E+02	6.226E+02	**6.007E+02**
	STD	5.514E+00	9.664E+00	1.120E+01	1.542E+01	5.486E+00	9.743E+00	3.159E+00	1.345E+01	1.381E+01	**1.118E+00**
F7	MEAN	1.071E+03	9.925E+02	1.121E+03	9.064E+02	1.284E+03	1.254E+03	8.627E+02	1.125E+03	9.738E+02	**8.496E+02**
	STD	6.536E+01	7.163E+01	8.817E+01	5.268E+01	7.367E+01	1.065E+02	4.135E+01	1.154E+02	5.619E+01	**4.521E+01**
F8	MEAN	9.509E+02	9.618E+02	9.945E+02	9.390E+02	9.779E+02	1.026E+03	1.088E+03	1.107E+03	9.211E+02	**9.006E+02**
	STD	2.795E+01	3.580E+01	2.837E+01	3.227E+01	2.698E+01	5.510E+01	3.277E+01	5.275E+01	1.808E+01	**2.387E+01**
F9	MEAN	6.113E+03	4.699E+03	5.744E+03	4.396E+03	7.605E+03	9.709E+03	1.935E+03	1.184E+04	3.275E+03	**1.102E+03**
	STD	9.379E+02	1.120E+03	7.999E+02	1.456E+03	9.218E+02	3.586E+03	6.568E+02	3.328E+03	5.496E+02	**3.215E+02**
F10	MEAN	5.617E+03	4.867E+03	5.832E+03	5.004E+03	5.769E+03	6.962E+03	4.662E+03	7.506E+03	4.991E+03	**4.485E+03**
	STD	6.331E+02	5.910E+02	6.399E+02	8.813E+02	6.189E+02	8.728E+02	1.163E+03	9.058E+02	2.824E+02	**3.525E+02**
F11	MEAN	1.715E+03	1.332E+03	2.187E+03	1.319E+03	1.273E+03	5.021E+03	1.702E+03	2.679E+03	1.206E+03	**1.194E+03**
	STD	2.496E+02	9.272E+01	9.252E+02	5.662E+01	4.743E+01	2.646E+03	5.609E+02	1.056E+03	2.665E+01	**3.224E+01**
F12	MEAN	4.042E+07	1.967E+06	1.290E+08	1.521E+07	2.088E+07	1.789E+08	8.350E+07	4.262E+08	4.402E+05	**4.292E+05**
	STD	2.575E+07	1.476E+06	1.652E+08	1.700E+07	1.585E+07	1.453E+08	9.517E+07	3.581E+08	3.580E+05	**3.174E+05**
F13	MEAN	4.988E+05	4.859E+04	8.658E+06	1.033E+05	5.288E+05	8.276E+05	4.032E+07	3.327E+07	1.662E+04	**8.535E+03**
	STD	3.207E+05	4.050E+04	3.021E+07	7.226E+04	2.134E+05	1.014E+06	1.073E+08	3.372E+07	1.312E+04	**8.976E+03**
F14	MEAN	4.984E+05	1.599E+04	4.518E+05	5.175E+04	3.596E+05	1.896E+06	6.574E+05	8.276E+05	6.250E+03	**6.214E+03**
	STD	5.448E+05	1.153E+04	7.153E+05	4.545E+04	3.386E+05	1.934E+06	1.114E+06	9.337E+05	5.011E+03	**5.789E+03**
F15	MEAN	9.090E+04	1.872E+04	6.949E+05	8.819E+04	5.990E+04	3.780E+05	1.250E+06	3.174E+05	3.299E+03	**3.194E+03**
	STD	5.337E+04	1.570E+04	1.371E+06	6.250E+04	3.181E+04	4.021E+05	4.525E+06	2.080E+05	2.299E+03	**1.584E+03**
F16	MEAN	3.256E+03	2.881E+03	3.152E+03	2.858E+03	3.398E+03	3.815E+03	2.500E+03	3.891E+03	2.527E+03	**2.449E+03**
	STD	4.271E+02	3.471E+02	4.293E+02	2.985E+02	3.895E+02	4.923E+02	3.278E+02	4.721E+02	1.816E+02	**3.494E+02**
F17	MEAN	2.292E+03	2.323E+03	2.311E+03	2.231E+03	2.582E+03	2.723E+03	1.950E+03	2.646E+03	1.922E+03	**1.859E+03**
	STD	2.894E+02	3.042E+02	2.027E+02	2.137E+02	2.807E+02	2.614E+02	1.250E+02	2.282E+02	6.039E+01	**9.363E+01**
F18	MEAN	2.573E+06	1.750E+05	2.239E+06	7.801E+05	1.870E+06	6.585E+06	1.210E+06	5.221E+06	1.224E+05	**1.135E+05**
	STD	2.104E+06	1.459E+05	2.238E+06	8.972E+05	2.640E+06	5.928E+06	1.809E+06	5.739E+06	7.009E+04	**1.110E+05**
F19	MEAN	1.548E+06	1.253E+04	6.131E+06	2.328E+06	6.954E+05	1.170E+07	7.106E+05	3.159E+07	5.024E+03	**4.456E+03**
	STD	1.010E+06	1.715E+04	2.479E+07	1.279E+06	5.597E+05	1.027E+07	1.118E+06	2.563E+07	2.666E+03	**1.927E+03**
F20	MEAN	2.583E+03	2.674E+03	2.690E+03	2.558E+03	2.802E+03	2.822E+03	2.434E+03	2.841E+03	2.360E+03	**2.283E+03**
	STD	1.930E+02	2.618E+02	1.528E+02	1.984E+02	1.980E+02	2.124E+02	1.774E+02	1.946E+02	8.031E+01	**9.139E+01**
F21	MEAN	2.469E+03	2.485E+03	2.515E+03	2.438E+03	2.565E+03	2.614E+03	2.386E+03	2.650E+03	2.417E+03	**2.367E+03**
	STD	2.666E+01	3.546E+01	3.803E+01	3.539E+01	4.495E+01	5.413E+01	1.960E+01	6.081E+01	1.658E+01	**1.664E+01**
F22	MEAN	2.414E+03	4.017E+03	3.466E+03	4.044E+03	6.522E+03	7.759E+03	4.985E+03	7.824E+03	2.301E+03	**2.300E+03**
	STD	8.520E+01	2.211E+03	1.171E+03	2.226E+03	1.875E+03	1.612E+03	2.046E+03	2.427E+03	2.055E+00	**8.666E−01**
F23	MEAN	2.917E+03	2.853E+03	2.924E+03	2.772E+03	3.197E+03	3.084E+03	2.769E+03	3.240E+03	2.764E+03	**2.724E+03**
	STD	5.389E+01	4.188E+01	6.594E+01	3.624E+01	1.274E+02	1.064E+02	5.021E+01	1.444E+02	2.353E+01	**2.061E+01**
F24	MEAN	3.059E+03	3.006E+03	3.064E+03	2.935E+03	3.442E+03	3.221E+03	2.916E+03	3.443E+03	2.927E+03	**2.885E+03**
	STD	5.831E+01	5.664E+01	6.381E+01	3.177E+01	1.394E+02	8.048E+01	4.362E+01	1.526E+02	2.270E+01	**1.658E+01**
F25	MEAN	2.953E+03	2.902E+03	3.070E+03	2.919E+03	2.929E+03	3.053E+03	2.984E+03	3.228E+03	2.906E+03	**2.889E+03**
	STD	2.940E+01	2.132E+01	8.566E+01	2.148E+01	2.145E+01	5.299E+01	5.647E+01	1.566E+02	1.608E+01	**8.287E+00**
F26	MEAN	5.164E+03	5.855E+03	6.052E+03	4.861E+03	7.108E+03	8.039E+03	4.632E+03	7.997E+03	**3.808E+03**	4.150E+03
	STD	1.675E+03	8.682E+02	1.099E+03	9.589E+02	1.863E+03	9.025E+02	2.541E+02	1.833E+03	**1.306E+03**	7.974E+02

(Continued)

Table 10 (continued)

Function	Index	AO	HPO	SCSO	SSA	BWO	HHO	DA	INGOfu	NGO	INGO
F27	MEAN	3.344E+03	3.245E+03	3.339E+03	3.260E+03	3.413E+03	3.402E+03	3.247E+03	3.477E+03	3.221E+03	**3.200E+03**
	STD	6.648E+01	2.218E+01	5.332E+01	3.630E+01	9.864E+01	1.016E+02	2.211E+01	1.372E+02	1.145E+01	**7.932E−05**
F28	MEAN	3.369E+03	**3.221E+03**	3.543E+03	3.258E+03	3.299E+03	3.461E+03	3.402E+03	3.759E+03	3.248E+03	3.300E+03
	STD	5.107E+01	**3.723E+01**	1.527E+02	4.097E+01	3.717E+01	9.048E+01	5.758E+01	2.857E+02	2.811E+01	5.118E−09
F29	MEAN	4.543E+03	4.079E+03	4.549E+03	4.186E+03	4.530E+03	5.099E+03	3.774E+03	5.126E+03	3.867E+03	**3.528E+03**
	STD	2.712E+02	3.034E+02	3.112E+02	2.273E+02	3.047E+02	4.559E+02	1.837E+02	4.247E+02	1.462E+02	**1.171E+02**
F30	MEAN	1.340E+07	3.413E+04	1.301E+07	6.554E+06	3.543E+06	2.764E+07	1.153E+07	4.237E+07	9.997E+03	**9.078E+03**
	STD	1.294E+07	2.582E+04	1.209E+07	6.005E+06	2.142E+06	2.169E+07	1.049E+07	3.184E+07	5.169E+03	**2.619E+03**

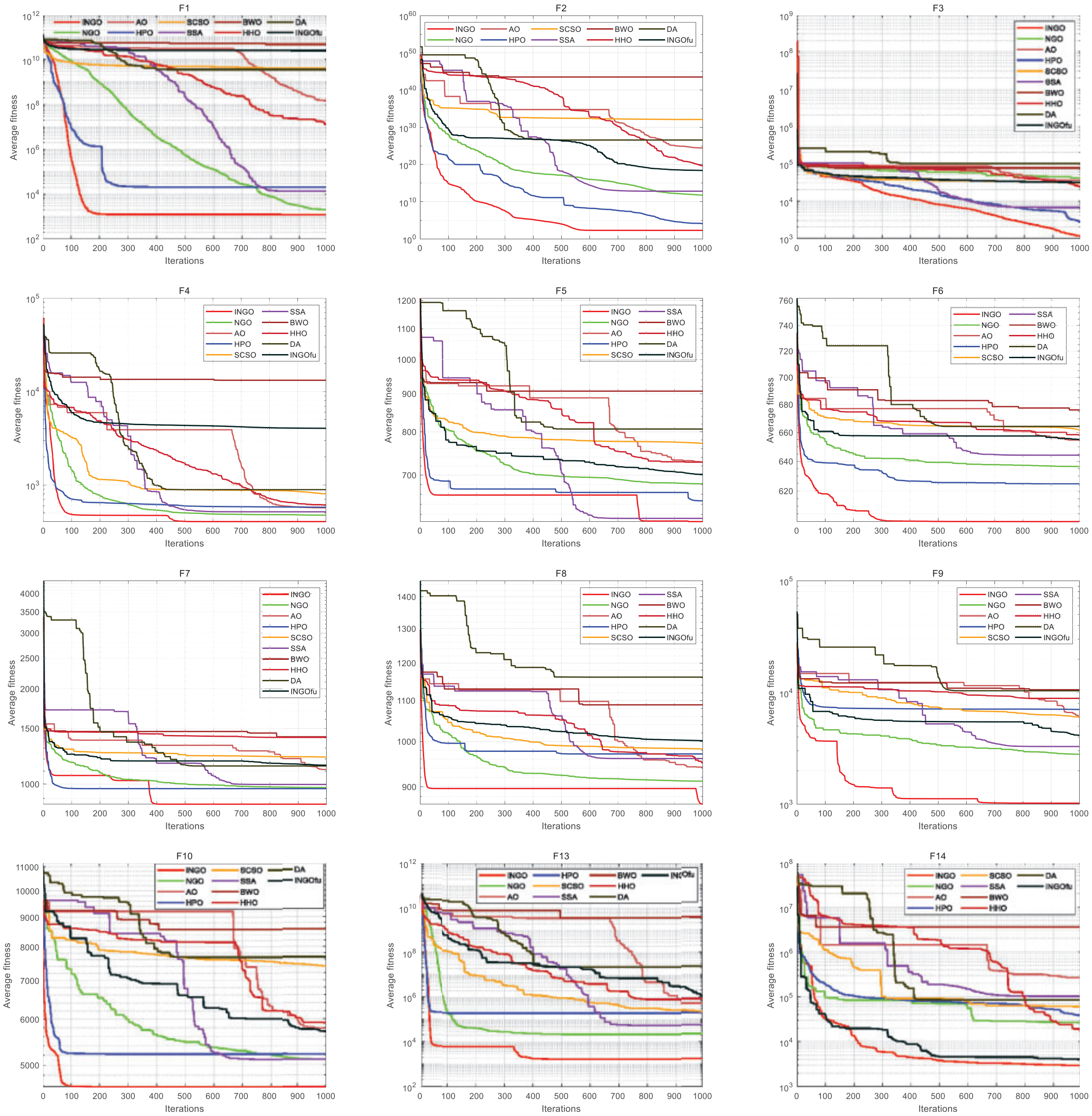

Figure 5: (Continued)

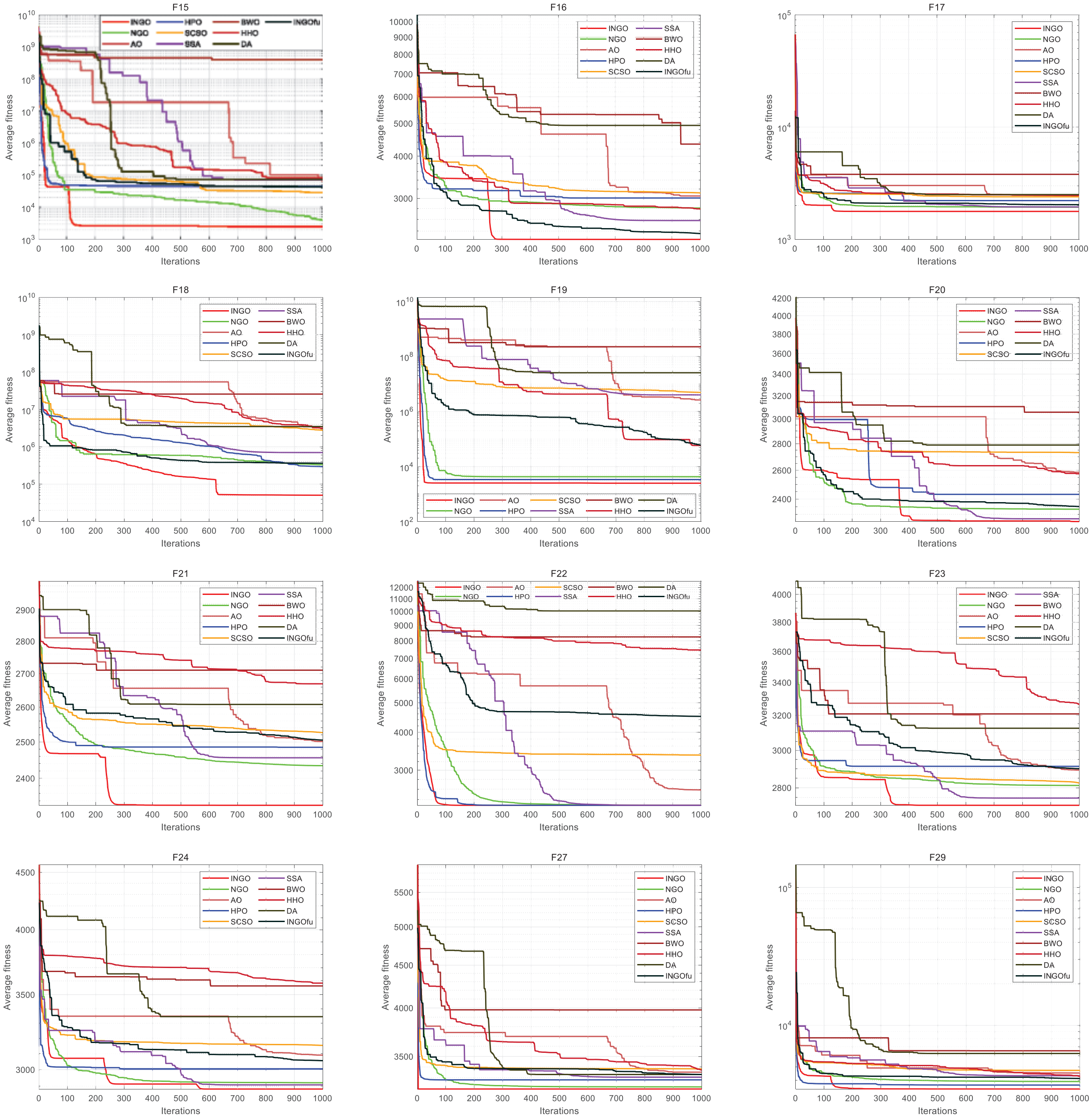

Figure 5: Iteration curve of CEC2017

Table 11 presents the statistical analysis outcomes of ten distinct algorithms evaluated on the CEC2017 test suite. Out of a total of 270 results, INGO achieved 249 instances of "+", 3 instances of "−", and 18 instances of "=", with the lowest AVR value of 1.20, securing the first position. This suggests that INGO has achieved more favorable optimization outcomes in comparison to the nine other algorithms evaluated.

Table 11: Wilcoxon rank-sum test on CEC2017 benchmark functions

INGO(VS)	INGO	NGO		AO		HPO		SCSO	
		P	S	P	S	P	S	P	S
F1	N/A	2.07E−02	+	3.02E−11	+	4.98E−04	+	3.02E−11	+
F2	N/A	3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
F3	N/A	3.02E−11	+	3.02E−11	+	**7.06E−01**	=	3.02E−11	+
F4	N/A	6.97E−03	+	3.02E−11	+	**4.04E−01**	=	4.50E−11	+
F5	N/A	6.72E−10	+	5.49E−11	+	2.15E−10	+	3.34E−11	+
F6	N/A	8.99E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
F7	N/A	5.07E−10	+	3.02E−11	+	1.69E−09	+	3.02E−11	+
F8	N/A	1.17E−03	+	1.20E−08	+	2.19E−08	+	8.15E−11	+
F9	N/A	3.34E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
F10	N/A	1.19E−06	+	7.12E−09	+	1.12E−02	+	5.07E−10	+
F11	N/A	**9.93E−02**	=	3.02E−11	+	4.18E−09	+	3.02E−11	+
F12	N/A	**8.30E−01**	=	3.02E−11	+	1.25E−07	+	3.02E−11	+
F13	N/A	2.62E−03	+	3.02E−11	+	2.67E−09	+	3.69E−11	+
F14	N/A	**6.00E−01**	=	4.08E−11	+	7.20E−05	+	8.99E−11	+
F15	N/A	**7.73E−01**	=	3.02E−11	+	2.19E−08	+	3.02E−11	+
F16	N/A	**2.23E−01**	=	1.43E−08	+	2.43E−05	+	7.09E−08	+
F17	N/A	7.70E−04	+	6.52E−09	+	9.26E−09	+	7.39E−11	+
F18	N/A	**2.58E−01**	=	6.07E−11	+	2.92E−02	+	1.09E−10	+
F19	N/A	**6.41E−01**	=	3.02E−11	+	**1.62E−01**	=	3.02E−11	+
F20	N/A	7.20E−05	+	4.57E−09	+	1.69E−09	+	4.50E−11	+
F21	N/A	8.15E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
F22	N/A	2.54E−08	+	2.92E−11	+	2.97E−04	+	2.92E−11	+
F23	N/A	7.69E−08	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
F24	N/A	1.10E−08	+	3.02E−11	+	3.34E−11	+	3.02E−11	+
F25	N/A	1.64E−05	+	1.09E−10	+	1.06E−03	+	3.02E−11	+
F26	N/A	**2.34E−01**	=	**2.23E−01**	=	3.82E−10	+	1.16E−07	+
F27	N/A	5.19E−10	+	2.79E−11	+	2.79E−11	+	2.79E−11	+
F28	N/A	1.21E−12	–	7.47E−10	+	1.21E−12	–	3.36E−11	+
F29	N/A	8.89E−10	+	3.02E−11	+	5.07E−10	+	3.02E−11	+
F30	N/A	**6.41E−01**	=	3.02E−11	+	1.09E−10	+	3.02E−11	+
+/−/=		21/0/9		29/0/1		26/1/3		30/0/0	
ARV	1.20	2.53		6.12		3.93		7.35	
RANK	1	2		6		3		8	
SSA		BWO		HHO		DA		INGOfu	
P	S	P	S	P	S	P	S	P	S
4.04E−01	=	3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
9.92E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
3.01E−04	+	1.41E−09	+	3.02E−11	+	4.08E−11	+	3.02E−11	+
5.53E−08	+	3.02E−11	+	3.02E−11	+	2.62E−03	+	3.02E−11	+

(Continued)

Table 11 (continued)

3.02E−11	+	3.02E−11	+	3.02E−11	+	7.39E−11	+	3.02E−11	+
1.32E−04	+	3.02E−11	+	3.02E−11	+	**3.48E−01**	=	3.02E−11	+
9.51E−06	+	8.99E−11	+	3.34E−11	+	3.02E−11	+	3.02E−11	+
6.70E−11	+	3.02E−11	+	3.02E−11	+	1.70E−08	+	3.02E−11	+
5.32E−03	+	2.87E−10	+	3.02E−11	+	**8.65E−01**	=	4.08E−11	+
2.15E−10	+	3.96E−08	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
3.02E−11	+	3.02E−1	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
8.15E−11	+	3.02E−11	+	3.02E−11	+	4.98E−11	+	3.02E−11	+
5.00E−09	+	6.07E−11	+	3.02E−11	+	2.15E−10	+	3.34E−11	+
3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
3.83E−05	+	3.82E−10	+	8.99E−11	+	**4.73E−01**	=	3.69E−11	+
1.86E−09	+	3.34E−11	+	3.02E−11	+	1.24E−03	+	3.02E−11	+
3.35E−08	+	9.76E−10	+	3.69E−11	+	2.23E−09	+	4.98E−11	+
3.02E−11	+	3.02E−11	+	3.02E−11	+	1.33E−10	+	3.02E−11	+
2.03E−07	+	3.34E−11	+	3.69E−11	+	1.17E−04	+	3.34E−11	+
1.21E−10	+	3.02E−11	+	3.02E−11	+	2.25E−04	+	3.02E−11	+
3.72E−09	+	2.92E−11	+	2.92E−11	+	2.92E−11	+	2.92E−11	+
3.81E−07	+	3.02E−11	+	3.02E−11	+	5.86E−06	+	3.02E−11	+
1.29E−09	+	3.02E−11	+	3.02E−11	+	2.39E−04	+	3.02E−11	+
1.60E−07	+	2.87E−10	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
8.66E−05	+	5.18E−07	+	3.02E−11	+	3.39E−02	+	4.68E−08	+
2.79E−11	+	2.79E−11	+	2.79E−11	+	2.79E−11	+	2.79E−11	+
1.33E−08	–	**6.41E−01**	=	1.21E−12	+	1.21E−12	–	1.21E−12	+
8.15E−11	+	3.02E−11	+	3.02E−11	+	2.78E−07	+	3.02E−11	+
3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+	3.02E−11	+
+/−/=	28/1/1	29/0/1		30/0/0		26/1/3		30/0/0	
ARV	4.17	6.47		8.73		4.95		9.55	
RANK	4	7		9		5		10	

5 Test of UAV Flight Path Planning Problem

5.1 The Introduction of the Problem

To verify INGO's performance in solving real-world problems, the algorithm is tested on a UAV flight path planning problem. UAV path planning represents a constrained optimization challenge where Swarm Intelligence (SI) finds widespread application [49–53].

Constrained optimization problems are typically addressed using the penalty function approach and the multi-objective approach [54]. The penalty function method has been widely used because of its simple execution. Its main idea is to construct the penalty adaptation function by incorporating penalty terms into the objective function and then solve this new unconstrained optimization problem.

UAV path planning is actually a multi-constraint combinatorial optimization problem [55], and constraints are dealt with in the form of penalty function, as shown in formula (20):

$$\begin{aligned} F(x) &= f(x) + \gamma \sum_{i=1}^{n} \delta_i^k(x) \cdot o(\delta_i(x)) \\ o(\delta_i(x)) &= \begin{cases} 1, \delta_i(x) > 0 \\ 0, \delta_i(x) \le 0 \end{cases} \end{aligned} \tag{20}$$

Here, $F(x)$ represents the objective function with penalty, $f(x)$ represents the original objective function, γ is the penalty parameter of the constraint condition. In addition, $k \in (1,2)$ and $\gamma \gg 1$ are the penalty parameters, $\delta_i(x)$ is the inequality constraint condition, n represents the number of inequality constraints.

To plan UAV flight path using Swarm Intelligent Optimization algorithm, the objective function should be determined first, and the total flight path evaluation function composed of the flight path length function, flight height function and angle cost function should be optimized. The total cost function is taken as the objective function of the algorithm. The indexes affecting UAV performance mainly include track length, flight altitude, minimum step length, corner cost, maximum climb angle, etc. [56]. Considering the three factors of track length, flight altitude and maximum turning angle, the total cost function is shown in formula (21):

$$\begin{cases} F(x) = w_1 f(L) + w_2 f(H) + w_3 f(J_{turn}) + \cdots \\ \sum_{i=1}^{n} w_i = 1 \end{cases} \tag{21}$$

Here, $F(x)$ is the total cost function, $f(L)$ is the cost function of the trajectory length, $f(H)$ is the cost function of the flight height, $f(J_{turn})$ is the cost function of the UAV turning angle, and w_i is the percentage of each cost function.

In the route planning problem processed by swarm intelligent algorithm, n nodes are selected from a complete route, where the distance between the i_{th} node and the $(i+1)_{th}$ node is denoted as $\|l(i+1) - l(i)\|_2$, and the coordinates of the two nodes are denoted as (x_i, y_i, z_i) and $(x_{i+1}, y_{i+1}, z_{i+1})$, labeled as $l(i)$ and $l(i+1)$, respectively. The mathematical model of track length of L_{path} is shown in formula (22):

$$L_{path} = \sum_{i=1}^{n-1} \|l(i+1) - l(i)\|_2 \tag{22}$$

The mathematical model of the standard deviation of flight altitude H_{height} is shown in formula (23), where H_{height} is the standard deviation cost function of altitude, $\bar{z}$ is the average altitude, and $z(i)$ is the height of the i_{th} node:

$$\begin{cases} H_{height} = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (z(i) - \bar{z})^2} \\ \bar{z} = \frac{1}{n} \sum_{i=0}^{n-1} z(i) \end{cases} \tag{23}$$

The mathematical model of turning angle J_{turn} is shown in formula (24), where Φ is the maximum turning angle, θ is the current turning angle, and a_i is the track vector of section i:

$$\begin{cases} \cos\theta = \dfrac{a_i^T a_{i+1}}{|a_i|\,|a_{i+1}|} \\ J_{turn} = \displaystyle\sum_{i=1}^{n} (\cos\Phi - \cos\theta_i) \end{cases} \tag{24}$$

5.2 *Analysis of Experimental Results*

To fully test the performance of the algorithm, the flight space is set to 250 ∗ 250 ∗ 60 km in the simulation and visualization process, and the coordinates of the starting and ending points are (10, 10, 10) and (210, 210, 20), respectively, and multiple peaks are added to the environment model to act as obstacles, with their parameters detailed in Table 12.

Table 12: The obstacles parameters

No.	h/km	X/km	Y/km	S_x/km	S_y/km
1	20	35	30	20	18
2	16	130	35	14	17
3	17	40	80	19	19
4	19	160	85	20	20
5	15	100	100	10	13
6	20	40	100	20	20
7	23	170	100	20	20
8	18	90	125	15	15
9	24	18	71	18	19
10	25	155	170	17	18
11	19	90	60	18	20
12	21	117	78	20	20
13	16	179	146	15	15

Under identical conditions, the INGO's outcomes are compared to those of NGO, SSA, BWO, DA and ChOA through 30 repetitions of the experiment. Table 13 displays the findings. The algorithm demonstrates its optimization ability and stability by achieving the minimum mean fitness value and standard deviation, with the optimal value of INGO being 217.2375 and the mean value being 224.1206. The results are superior when compared to other algorithms. It is noteworthy that, in contrast to other foundational algorithms such as SSA and BWO, the NGO algorithm does not exhibit rapid characteristics in the context of UAV trajectory planning problems under this mathematical model, despite its capability to achieve higher precision. The enhanced INGO algorithm, compared to the original NGO algorithm, demonstrates improvements in both speed and accuracy for such problems, validating the effectiveness of our four strategies for algorithmic enhancement. However, within the framework of the NGO algorithm, even the speed performance of INGO remains inferior compared to other algorithms. Therefore, at this stage, INGO is more apt for addressing static UAV trajectory planning problems that disregard temporal influences but demand cost minimization.

Additionally, exploring ways to enhance INGO's speed performance in these problems constitutes a potential direction for our future considerations.

Table 13: UAV flight path planning results

No.	Indicators	INGO	NGO	SSA	BWO	DA	ChOA
1	Optimal	217.2375	231.1058	220.5907	248.8895	**215.0465**	216.9916
2	Worst	**238.5341**	257.9022	246.7527	368.2322	286.4377	319.5354
3	Mean	**224.1206**	241.8038	242.7427	285.0163	234.9963	246.0827
4	STD	**5.9637**	7.4185	32.6120	34.4144	15.7019	24.1947
5	Time (s)	12.8619	85.0672	7.8375	3.0306	4.5729	**1.100**

Fig. 6 shows the convergence curves of the six algorithms. Compared with other algorithms, INGO gets a smaller fitness value. Compared to NGO, which can approximate the optimal solution in nearly 10 iterations, it has a faster rate of convergence.

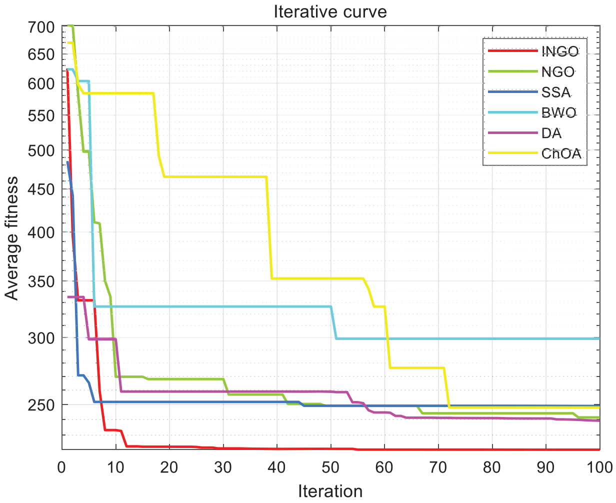

Figure 6: Fitness values change with the iteration

Figs. 7 and 8 show that swarm intelligence algorithm can plan a reasonable flight path for UAV in the case of obstacles and complete the goal of flight path planning. And the planned flight path is at a more stable and suitable altitude. The above experimental results show that INGO has good performance in this kind of problem and can effectively solve the UAV flight path planning problem under this environmental model.

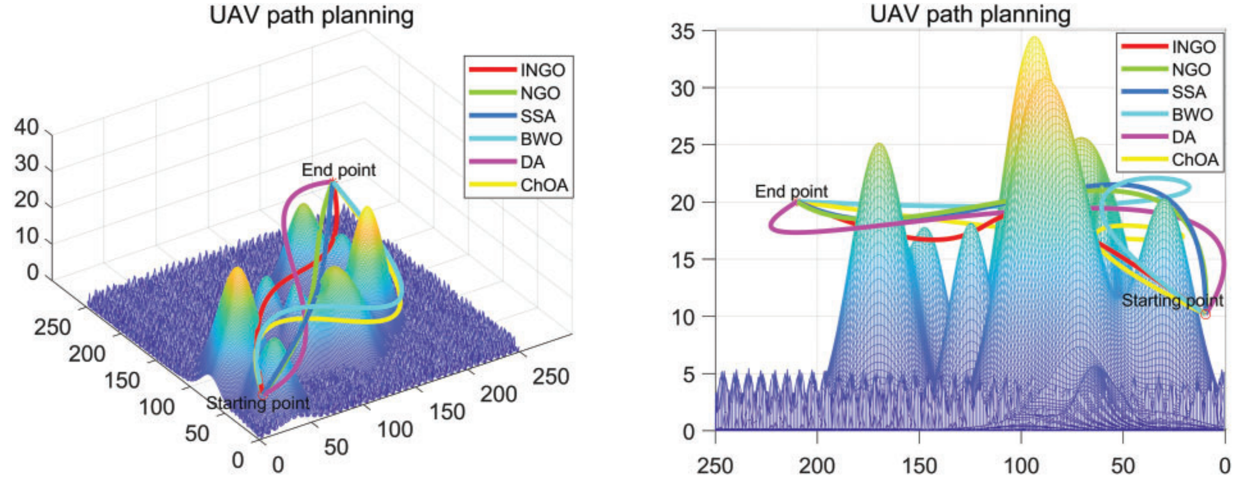

Figure 7: UAV path planning with different algorithms

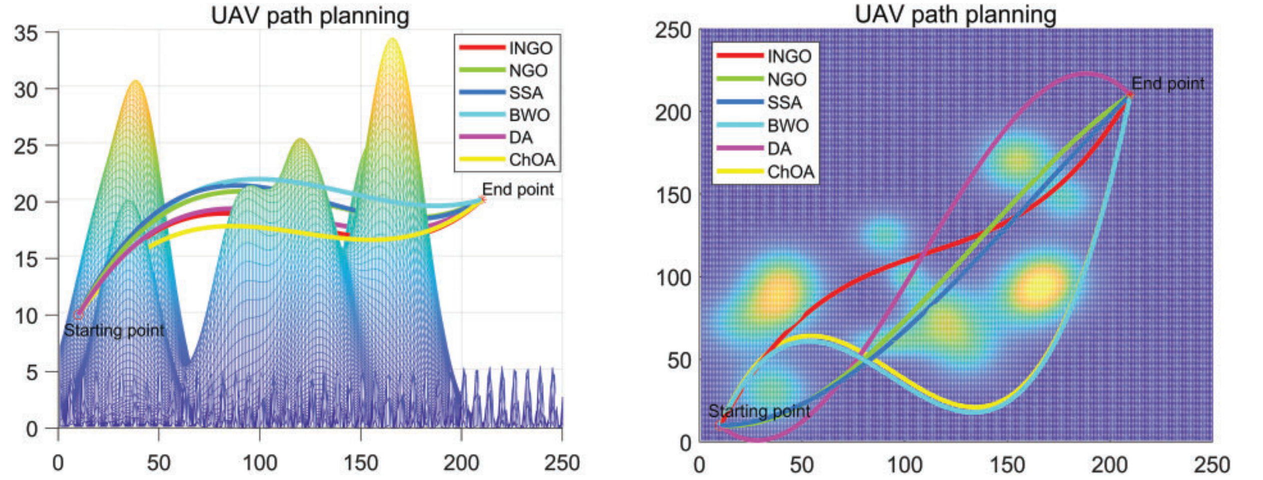

Figure 8: Front view and top view of UAV path planning

6 Conclusion

Within the realm of function optimization, the NGO algorithm exhibits diminished convergence accuracy and a propensity for entrapment in local optima. This paper proposes four strategic enhancements aimed at harmonizing the exploratory and exploitative abilities of NGO. First, introducing tent chaotic mapping aims to uniformly distribute the initial population, increasing diversity. Next, incorporating whale fall phenomena and T-distribution variations increases the algorithm's chance of escaping local optima, reducing the risk of falling into local minima. Lastly, integrating the state transition algorithm's neighborhood search capability strengthens INGO's local utilization. INGO's performance, assessed on public test sets and UAV trajectory planning, was compared with classic and new algorithms. Statistical methods analyzed the experimental results, summarizing the algorithm's strengths and areas for improvement. The findings show that INGO has higher convergence accuracy and a better ability to escape local extremes in function optimization.

Moving forward, our efforts will focus on three main avenues: Firstly, we aim to integrate various optimization strategies into advanced intelligent optimization algorithms, creating competitive solutions for real-world problems. Secondly, considering INGO's limitations, we will explore the inclusion of adaptive adjustment strategies to enhance algorithmic performance. Lastly, we will extend the application of INGO to tackle more sophisticated engineering scenarios, such as the complex task of UAV trajectory planning that involves additional constraints, broadening our approach to encompass a wider spectrum of intricate engineering problems.

Acknowledgement: The authors wish to acknowledge the editor and anonymous reviewers for their insightful comments, which have improved the quality of this publication.

Funding Statement: This work was in part supported by the Key Research and Development Project of Hubei Province (No. 2023BAB094), the Key Project of Science and Technology Research Program of Hubei Educational Committee (No. D20211402), and the Open Foundation of Hubei Key Laboratory for High-Efficiency Utilization of Solar Energy and Operation Control of Energy Storage System (No. HBSEES202309).

Author Contributions: Study conception and design: Mai Hu, Liang Zeng; Data collection: Chenning Zhang, Quan Yuan and Shanshan Wang; Analysis and interpretation of results: All authors; Draft manuscript preparation: Mai Hu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data and materials are available.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] L. Abualigah *et al.*, "Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: A comprehensive survey, applications, comparative analysis, and results," *Neural Comput. Appl.*, vol. 34, no. 6, pp. 1–30, 2022. doi: 10.1007/s00521-021-06747-4.

[2] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Comput. Commun.*, vol. 149, no. 12, pp. 270–299, 2020. doi: 10.1016/j.comcom.2019.10.014.

[3] L. L. Li, Z. F. Liu, M. L. Tseng, S. J. Zheng, and M. K. Lim, "Improved tunicate swarm algorithm: Solving the dynamic economic emission dispatch problems," *Appl. Soft Comput.*, vol. 108, no. 5, pp. 107504, 2021. doi: 10.1016/j.asoc.2021.107504.

[4] E. H. Houssein, A. G. Gad, Y. M. Wazery, and P. N. Suganthan, "Task scheduling in cloud computing based on meta-heuristics: Review, taxonomy, open challenges, and future trends," *Swarm Evol. Comput.*, vol. 62, no. 3, pp. 100841, 2021. doi: 10.1016/j.swevo.2021.100841.

[5] D. Zhao *et al.*, "Ant colony optimization with horizontal and vertical crossover search: Fundamental visions for multi-threshold image segmentation," *Expert. Syst. Appl.*, vol. 167, no. 1, pp. 114122, 2021. doi: 10.1016/j.eswa.2020.114122.

[6] Z. Sadeghian, E. Akbari, H. Nematzadeh, and H. Motameni, "A review of feature selection methods based on meta-heuristic algorithms," *J. Exp. Theor. Artif. Intell.*, vol. 78, no. 2, pp. 1–51, 2023. doi: 10.1080/0952813X.2023.2183267.

[7] S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks*, Springer, Cham, 2019, vol. 780, pp. 43–55. doi: 10.1007/978-3-319-93025-1_4.

[8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN'95-Int. Conf. on Neural Netw.*, Perth, WA, Australia, IEEE, 1995, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.

[9] R. A. Rutenbar, "Simulated annealing algorithms: An overview," *IEEE Circuits Devices Mag.*, vol. 5, no. 1, pp. 19–26, 1989. doi: 10.1109/101.17235.

[10] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl. Based Syst.*, vol. 96, no. 63, pp. 120–133, 2016. doi: 10.1016/j.knosys.2015.12.022.

[11] A. R. Jordehi, "Brainstorm optimisation algorithm (BSOA): An efficient algorithm for finding optimal location and setting of FACTS devices in electric power systems," *Int. J. Electr. Power Energy Syst.*, vol. 69, no. 10, pp. 48–57, 2015. doi: 10.1016/j.ijepes.2014.12.083.

[12] F. Glover, "Tabu search: A tutorial," *Interfaces*, vol. 20, no. 4, pp. 74–94, 1990. doi: 10.1287/inte.20.4.74.

[13] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, 2006. doi: 10.1109/MCI.2006.329691.

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

[15] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, no. 12, pp. 51–67, 2016. doi: 10.1016/j.advengsoft.2016.01.008.

[16] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, 2016. doi: 10.1007/s00521-015-1920-1.

[17] L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. Al-Qaness and A. H. Gandomi, "Aquila optimizer: A novel meta-heuristic optimization algorithm," *Comput. Ind. Eng.*, vol. 157, no. 11, pp. 107250, 2021. doi: 10.1016/j.cie.2021.107250.

[18] A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, 2019. doi: 10.1016/j.future.2019.02.028.

[19] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert. Syst. Appl.*, vol. 177, no. 8, pp. 114864, 2021. doi: 10.1016/j.eswa.2021.114864.

[20] A. E. Ezugwu, J. O. Agushaka, L. Abualigah, S. Mirjalili, and A. H. Gandomi, "Prairie dog optimization algorithm," *Neural Comput. Appl.*, vol. 34, no. 22, pp. 20017–20065, 2022. doi: 10.1007/s00521-022-07530-9.

[21] J. O. Agushaka, A. E. Ezugwu, and L. Abualigah, "Dwarf mongoose optimization algorithm," *Comput. Methods Appl. Mech. Eng.*, vol. 391, pp. 114570, 2022.

[22] M. Khishe and M. R. Mosavi, "Chimp optimization algorithm," *Expert. Syst. Appl.*, vol. 149, no. 1, pp. 113338, 2020. doi: 10.1016/j.eswa.2020.113338.

[23] C. Zhong, G. Li, and Z. Meng, "Beluga whale optimization: A novel nature-inspired metaheuristic algorithm," *Knowl. Based Syst.*, vol. 251, no. 1, pp. 109215, 2022. doi: 10.1016/j.knosys.2022.109215.

[24] M. Dehghani, Š. Hubálovský, and P. Trojovský, "Northern goshawk optimization: A new swarm-based algorithm for solving optimization problems," *IEEE Access*, vol. 9, pp. 162059–162080, 2021. doi: 10.1109/ACCESS.2021.3133286.

[25] S. Yu and J. Ma, "Adaptive composite fault diagnosis of rolling bearings based on the CLNGO algorithm," *Processes*, vol. 10, no. 12, pp. 2532, 2022. doi: 10.3390/pr10122532.

[26] H. Youssef, S. Kamel, M. H. Hassan, J. Yu, and M. Safaraliev, "A smart home energy management approach incorporating an enhanced northern goshawk optimizer to enhance user comfort, minimize costs, and promote efficient energy consumption," *Int. J. Hydrogen Energy*, vol. 49, no. 57, pp. 644–658, 2024. doi: 10.1016/j.ijhydene.2023.10.174.

[27] Y. Liang, X. Hu, G. Hu, and W. Dou, "An enhanced northern goshawk optimization algorithm and its application in practical optimization problems," *Mathematics*, vol. 10, no. 22, pp. 4383, 2022. doi: 10.3390/math10224383.

[28] M. A. El-Dabah, R. A. El-Sehiemy, H. M. Hasanien, and B. Saad, “Photovoltaic model parameters identification using northern goshawk optimization algorithm,” *Energy*, vol. 262, pp. 125522, 2023. doi: 10.1016/j.energy.2022.125522.

[29] H. Satria, R. B. Syah, M. L. Nehdi, M. K. Almustafa, and A. O. I. Adam, “Parameters identification of solar PV using hybrid chaotic northern goshawk and pattern search,” *Sustainability*, vol. 15, no. 6, pp. 5027, 2023. doi: 10.3390/su15065027.

[30] J. Zhong, T. Chen, and L. Yi, “Face expression recognition based on NGO-BILSTM model,” *Front. Neurorobot.*, vol. 17, pp. 1155038, 2023. doi: 10.3389/fnbot.2023.1155038.

[31] G. Tian *et al.*, “Hybrid evolutionary algorithm for stochastic multiobjective disassembly line balancing problem in remanufacturing,” *Environ. Sci. Pollut. Res.*, vol. 55, no. 24, pp. 1–16, 2023. doi: 10.1007/s11356-023-27081-3.

[32] J. Wang, Z. Xiang, X. Cheng, J. Zhou, and W. Li, “Tool wear state identification based on SVM optimized by the improved northern goshawk optimization,” *Sensors*, vol. 23, no. 20, pp. 8591, 2023. doi: 10.3390/s23208591.

[33] C. Zhan *et al.*, “Environment-oriented disassembly planning for end-of-life vehicle batteries based on an improved northern goshawk optimisation algorithm,” *Environ. Sci. Pollut. Res.*, vol. 30, no. 16, pp. 47956–47971, 2023. doi: 10.1007/s11356-023-25599-0.

[34] J. Bian, R. Huo, Y. Zhong, and Z. Guo, “XGB-Northern goshawk optimization: Predicting the compressive strength of self-compacting concrete,” *KSCE J. Civil Eng.*, vol. 28, no. 4, pp. 1–17, 2024. doi: 10.1007/s12205-024-1647-6.

[35] X. Wang, J. Wei, F. Wen, and K. Wang, “A trading mode based on the management of residual electric energy in electric vehicles,” *Energies*, vol. 16, no. 17, pp. 6317, 2023. doi: 10.3390/en16176317.

[36] M. Ning *et al.*, “Revealing the hot deformation behavior of AZ42 Mg alloy by using 3D hot processing map based on a novel NGO-ANN model,” *J. Mater. Res. Technol.*, vol. 27, pp. 2292–2310, 2023. doi: 10.1016/j.jmrt.2023.10.073.

[37] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997. doi: 10.1109/4235.585893.

[38] S. Gupta and K. Deep, “A novel random walk grey wolf optimizer,” *Swarm Evol. Comput.*, vol. 44, no. 4, pp. 101–112, 2019. doi: 10.1016/j.swevo.2018.01.001.

[39] X. Zhou, C. Yang, and W. Gui, “State transition algorithm,” arXiv preprint arXiv:1205.6548, 2012.

[40] A. Tharwat and W. Schenck, “Population initialization techniques for evolutionary algorithms for single-objective constrained optimization problems: Deterministic vs. stochastic techniques,” *Swarm Evol. Comput.*, vol. 67, pp. 100952, 2021. doi: 10.1016/j.swevo.2021.100952.

[41] Y. Li, M. Han, and Q. Guo, “Modified whale optimization algorithm based on tent chaotic mapping and its application in structural optimization,” *KSCE J. Civil Eng.*, vol. 24, no. 12, pp. 3703–3713, 2020. doi: 10.1007/s12205-020-0504-5.

[42] J. H. Liu and Z. H. Wang, “A hybrid sparrow search algorithm based on constructing similarity,” *IEEE Access*, vol. 9, pp. 117581–117595, 2021. doi: 10.1109/ACCESS.2021.3106269.

[43] P. N. Suganthan *et al.*, “Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization,” *KanGAL Rep.*, 2005.

[44] G. Wu, R. Mallipeddi, and P. N. Suganthan, “Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization,” *Tech. Rep,* National University of Defense Technology, Changsha, Hunan, China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, 2017.

[45] I. Naruei, F. Keynia, and A. Sabbagh Molahosseini, “Hunter-prey optimization: Algorithm and applications,” *Soft Comput.*, vol. 26, no. 3, pp. 1279–1314, 2022. doi: 10.1007/s00500-021-06401-0.

[46] Seyyedabbasi and F. Kiani, “Sand cat swarm optimization: A nature-inspired algorithm to solve global optimization problems,” *Eng. Comput.*, vol. 39, no. 4, pp. 2627–2651, 2023. doi: 10.1007/s00366-022-01604-x.

[47] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris and S. M. Mirjalili, “Salp swarm algorithm: A bio-inspired optimizer for engineering design problems,” *Adv. Eng. Softw.*, vol. 114, pp. 163–191, 2017. doi: 10.1016/j.advengsoft.2017.07.002.

[48] X. Fu, L. Zhu, J. Huang, J. Wang, and A. Ryspayev, “Multi-threshold image segmentation based on the improved northern goshawk optimization,” *Comput. Eng.*, vol. 49, no. 7, pp. 232–241, 2023.

[49] H. S. Yahia and A. S. Mohammed, “Path planning optimization in unmanned aerial vehicles using meta-heuristic algorithms: A systematic review,” *Environ. Monit. Assess.*, vol. 195, no. 1, pp. 30, 2023. doi: 10.1007/s10661-022-10590-y.

[50] Y. Wu, “A survey on population-based meta-heuristic algorithms for motion planning of aircraft,” *Swarm Evol. Comput.*, vol. 62, no. 6, pp. 100844, 2021. doi: 10.1016/j.swevo.2021.100844.

[51] J. Tang, H. Duan, and S. Lao, “Swarm intelligence algorithms for multiple unmanned aerial vehicles collaboration: A comprehensive review,” *Artif. Intell. Rev.*, vol. 56, no. 5, pp. 4295–4327, 2023. doi: 10.1007/s10462-022-10281-7.

[52] R. A. Saeed, M. Omri, S. Abdel-Khalek, E. S. Ali, and M. F. Alotaibi, “Optimal path planning for drones based on swarm intelligence algorithm,” *Neural Comput. Appl.*, vol. 34, no. 12, pp. 10133–10155, 2022. doi: 10.1007/s00521-022-06998-9.

[53] Yao *et al.*, “IHSSAO: An improved hybrid salp swarm algorithm and aquila optimizer for UAV path planning in complex terrain,” *Appl. Sci.*, vol. 12, no. 11, pp. 5634, 2022. doi: 10.3390/app12115634.

[54] G. Guo and Y. Wang, “Constrained-handling techniques based on evolutionary algorithms,” *J. Hunan Univ. (Nat. Sci.)*, vol. 19, no. 4, pp. 15–18, 2006.

[55] Y. Chen, D. Pi, and Y. Xu, “Neighborhood global learning based flower pollination algorithm and its application to unmanned aerial vehicle path planning,” *Expert. Syst. Appl.*, vol. 170, pp. 114505, 2021. doi: 10.1016/j.eswa.2020.114505.

[56] Z. Yu, Z. Si, X. Li, D. Wang, and H. Song, “A novel hybrid particle swarm optimization algorithm for path planning of UAVs,” *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22547–22558, 2022. doi: 10.1109/JIOT.2022.3182798.