

Multi-strategy enhanced Aquila optimization algorithm

Guo Hairu, Zhao Longfei (✉), Liu Yongli

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454000, China

Abstract

Although the Aquila optimization (AO) algorithm demonstrates promising results in various optimization tasks, it is still easy to fall into local optima, particularly when addressing more intricate problems. To tackle this issue, a reinforced tent-enhanced AO (RTEAO) algorithm that incorporates multiple strategic integrations was proposed in this paper. During the initialization phase, a tent chaotic mapping method is employed to enhance population diversity. To update individual positions iteratively, an elite reverse learning strategy and an optimal retention mechanism are employed to circumvent local optima. Experimental evaluations on the CEC2005 test functions demonstrate significant improvements in both convergence speed and accuracy. Furthermore, the RTEAO algorithm showcases remarkable optimization performance in practical applications, specifically in the calibration of a back propagation (BP) neural network for regression analysis on the Boston Housing Price Dataset and Wisconsin Breast Cancer Dataset, thereby affirming its efficacy and versatility in confronting real-world optimization challenges.

Keywords Aquila optimization (AO), reinforced tent-enhanced AO (RTEAO), tent chaotic mapping, elite reverse learning, optimal retention mechanism, back propagation (BP) neural network

1 Introduction

As real-world optimization problems become more complex, there is a growing need for efficient computational methods to find solutions. Metaheuristic optimization algorithms emerge as effective tools that mimic natural phenomena or biological behaviors in order to identify optimal solutions. These approaches demonstrate exceptional optimization capabilities across various fields, attracting considerable research interest.

Current metaheuristic algorithms are generally

classified into two main categories: evolutionary and swarm intelligence methods, and physics-based methods.

Evolutionary and swarm intelligence methods are inspired by biological evolution and collective behaviors. These approaches explore the solution space using population-based stochastic search. In each generation, the fittest solutions guide subsequent evolutionary improvements. Notable examples include genetic algorithms (GA)^[1-2] and the grey wolf optimizer (GWO)^[3]. The whale optimization algorithm (WOA)^[4] simulates the unique ‘bubble net predation strategy’ of humpback whales to tackle optimization problems. The ant colony optimization (ACO)^[5] mimics the pheromone traces left by ants during foraging, allowing them to efficiently search for and find solutions through communication via these

pheromones. Physics-based methods draw inspiration from fundamental physical phenomena and their governing principles to solve complex optimization challenges. A classic example is the simulated annealing (SA)^[6] algorithm, which effectively avoids local optima by simulating the cooling process in metal annealing. This approach uses a temperature parameter to control the acceptance criterion during the search process, gradually steering toward the global optimum. Another example is the frequencies wave sound particle swarm optimization (FPSO), which is inspired by the propagation and interaction of sound waves in nature. FPSO integrates particle swarm optimization (PSO) principles with the characteristics of sound wave propagation and reflection to identify optimal solutions.

The performance of metaheuristic optimization algorithms differs greatly across various application domains. While an algorithm may perform exceptionally well in one specific area, this does not ensure similar results in other problem spaces. Both evolutionary swarm intelligence methods and physics-based approaches demonstrate promising optimization capabilities for standard problems, however, their effectiveness often decreases when dealing with high-dimensional and complex optimization challenges.

To address these limitations, Abualigah et al.^[7] proposed the AO algorithm. It offers two main advantages for multidimensional problems: superior convergence efficiency and enhanced solution accuracy. However, this rapid convergence characteristic can also increase the risk of premature convergence to local optima.

The inherent trade-off between convergence speed and global search capability highlights the need for careful algorithm design and parameter tuning, especially in complex, multimodal optimization landscapes. Consequently, enhancing the robustness and overall performance of the AO algorithm has become a significant research focus in recent years. Several notable improvements have been proposed. Wang et al.^[8] significantly improved the performance of the algorithm by integrating AO with the Harris Hawks optimization (HHO)^[9]. This combination introduced novel hunting strategies and adversarial

learning mechanisms, enhancing the algorithm's convergence behavior and global search capability. Fatani et al.^[10] demonstrated the effectiveness of AO in image processing through their hybrid MobileNetV3-AO architecture, achieving state-of-the-art results on benchmark image classification datasets. Zhang et al.^[11] successfully expanded AO's applicability by combining it with the arithmetic optimization algorithm (AOA)^[12], particularly for addressing high-dimensional optimization problems. In the field of renewable energy systems, El Marzougui et al.^[13] achieved superior parameter extraction accuracy using a chaos-enhanced adaptive Jaya (a Sanskrit word meaning victory) algorithm, showcasing the potential for metaheuristics in energy applications. Wu et al.^[14] proposed the parallel and self-adaptive Aquila optimizer (PSAO), which incorporates three key innovations. It utilizes a space-filling good point set for superior initial population distribution, integrates a golden sine operator with particle swarm-inspired self-learning and social learning strategies, and implements a nonlinear equilibrium factor as the algorithm's switching criterion. To further refine the balance between exploration and improve search precision, Zhang et al.^[15] developed the niche centroid adaptive AO (NCAAO).

These enhanced versions of AO show how hybrid strategies and adaptive mechanisms can effectively tackle the algorithm's inherent challenges in maintaining optimal search behavior during various optimization phases.

Although the algorithms mentioned above achieve some success, the problem of easily falling into local optima still exists. The RTEAO algorithm is proposed, which introduces a tent chaotic mapping^[16–18] in the initialization phase to enhance the diversity and distribution uniformity of initial solutions. During the update of the global optimal value, the elite reverse learning strategy^[19–21] is used to calculate the elite reverse values and is applied to the individual position iteration update stage together with the current optimal value. This approach helps prevent the algorithm from falling into local optima and ensures stability and convergence accuracy throughout the search process.

Additionally, the algorithm features an optimal retention mechanism^[22–24]. By comparing the positions of two candidate individuals obtained during the population update phase, the location with the best fitness value is selected for the individual update. This ensures that the algorithm progressively moves closer to the global optimal solution during the iterative process.

2 AO algorithm

The design inspiration of AO algorithm comes from the fact that Aquila can switch between different hunting strategies when facing different situations. The position of the Aquila in the D -dimensional search space represents a set of effective solutions, and the energy storage of the Aquila represents the fitness value. The total number of candidate solutions is N . X represents a group of current candidate solutions, which are randomly generated by

$$X_{n,d} = B_d^l + r(B_d^u - B_d^l); \quad n = 1, 2, \dots, N, \quad d = 1, 2, \dots, D \quad (1)$$

where $X_{n,d}$ denotes the position of the n th Aquila individual in the d th dimension, r is a random number uniformly distributed in the interval $(0, 1)$, B_d^u and B_d^l are the upper and lower bounds of the d th dimension of the search space, respectively, N is the population size, and D is the dimensionality of the problem.

2.1 First stage

In this stage, after selecting the best hunting position, the Aquila will fly high and dive vertically over the determined prey area. Aquila flies over the high-altitude hunting area to find the best location for prey. Once they find the best location, they will dive vertically to catch prey. This behavior can be described by

$$X_1(t+1) = X_b(t) \left(1 - \frac{t}{T}\right) + (\bar{X}(t) - rX_b(t)) \quad (2)$$

The solution of the $(t+1)$ th iteration generated by this method is represented by $X_1(t+1)$. The optimal solution of the t th iteration is expressed as $X_b(t)$, which is approximately the location of prey. t and T represent the current iteration and the maximum number of iterations, respectively. $\bar{X}(t)$ represents the

mean value of the current solution at iteration t . This behavior can be described by

$$\bar{X}(t) = \frac{1}{N} \sum_{n=1}^N X_n(t) \quad (3)$$

2.2 Second stage

At this stage, the Aquila will switch from flying at high-altitude to hovering around the head of the prey to prepare for the attack. Then, attack from high-altitude when the prey's position is determined. This behavior can be described by

$$X_2(t+1) = X_b(t)F_{LF}(D) + X_R(t) + r(y - x) \quad (4)$$

where $X_2(t+1)$ denotes the solution obtained at the $(t+1)$ th iteration by the second strategy, x and y denote the coordinate parameters of the spiral flight trajectory in the position search process, and $X_R(t)$ represents a randomly selected solution at the t th iteration. The Lévy flight distribution function is defined as

$$F_{LF}(D) = s \frac{\mu\sigma}{|v|^{1/\beta}} \quad (5)$$

where μ and v are random variables drawn from normal distributions in the interval $(0, 1)$, and σ is calculated as

$$\sigma = \left(\frac{\Gamma(1+\beta) \sin \frac{\pi\beta}{2}}{2^{\frac{\beta-1}{2}} \beta \Gamma\left(\frac{1+\beta}{2}\right)} \right)^{\frac{1}{\beta}} \quad (6)$$

where $\Gamma(\cdot)$ is the Gamma function and β is the key distribution parameter of the Lévy flight, μ is a random variable controlling the step length, and s denotes the step length traveled by an individual during one Lévy flight.

2.3 Third stage

The Aquila will find and determine the approximate position of the prey through the previous stage, and after finding the prey, it will plunge vertically in pursuit of the prey. This behavior is shown as

$$X_3(t+1) = X_b(t) - \alpha\bar{X}(t) - r + \delta((B^u - B^l)r + B^l) \quad (7)$$

where $X_3(t+1)$ denotes the solution obtained at the $(t+1)$ th iteration in this stage, α and δ are adjustment parameters controlling the exploitation

process and are both set to 0.1, r is a random number uniformly distributed in the interval $(0, 1)$, and B^u and B^l represent the upper and lower bounds of the search space, respectively.

2.4 Fourth stage

In this stage, the Aquila will follow the escape track of the prey to land and chase. When the prey is within the attack distance, the Aquila will follow the random movement of the prey to attack. This behavior is shown as

$$X_4(t+1) = F_Q(t)X_b(t) - rG_1X(t) - G_2F_{LF}(D) + rG_1 \quad (8)$$

where $X_4(t+1)$ denotes the solution obtained at the $(t+1)$ th iteration in this stage, $X(t)$ represents the current solution at the t th iteration, and $F_Q(t)$ is the mass function used to balance exploration and exploitation, which is calculated as

$$F_Q(t) = t^{\frac{2r-1}{(1-r)^2}} \quad (9)$$

$$G_1 = 2r - 1 \quad (10)$$

$$G_2 = 2 \left(1 - \frac{t}{T}\right) \quad (11)$$

where G_1 and G_2 are control coefficients that regulate the movement behavior, r is a random number uniformly distributed in the interval $(0, 1)$, t denotes the current iteration index, and T is the maximum number of iterations.

3 RTEAO algorithm

Based on the limitations of the AO, a multi-strategy enhanced AO algorithm, RTEAO, is introduced. The RTEAO algorithm incorporates a tent chaotic mapping, an elite opposition-based learning strategy, and an optimal retention mechanism. These enhancements are designed to comprehensively boost the algorithm's optimization performance, ensuring more efficient and effective solutions.

3.1 Tent chaotic mapping

The diversity and uniformity of the initial distribution of population particles play a critical role in determining the global search capability and convergence efficiency of an algorithm. To tackle this

issue, tent chaotic mapping is introduced during the population initialization process. The RTEAO algorithm ensures a more uniform and dispersed distribution of the initial population, thereby enhancing the algorithm's exploration ability. The tent chaotic mapping is mathematically described as

$$x_{n+1} = \begin{cases} 2x_n; & 0 \leq x_n \leq 0.5 \\ 2(1-x_n); & 0.5 < x_n \leq 1.0 \end{cases} \quad (12)$$

where x_n is the position of the n th Aquila individual, x_{n+1} denotes the position of the $(n+1)$ th Aquila individual.

3.2 Elite reverse learning

The individual update position in the RTEAO algorithm heavily relies on the optimal value. This dependency poses a risk: if the algorithm converges to a local optimum, the entire population might also become trapped in this suboptimal state. Given the greedy selection process, where each subsequent iteration tends to favor positions with better fitness values, this can lead to a rapid decrease in population diversity and premature convergence.

To counteract these issues, an elite reverse learning strategy is incorporated into the optimal value update process. By employing this strategy, the algorithm calculates a position opposite to the current optimal value using the elite reverse learning formula. Introducing this opposite position into the iterative update mechanism of individual positions helps maintain diversity within the population. It assists in escaping from local optima, thus enhancing the overall search capability and preventing premature convergence. This strategy calculates the position X_F opposite to the optimal value through the elite reverse learning formula, and substitutes X_F into the calculation of the iterative update of the individual position of the population. Finally, the middle value of the two new positions is taken as the updated individual position, so as to effectively jump out of the local optimum. The calculation of elite reverse learning is shown as

$$X_F = k(B^u + B^l) - X_b \quad (13)$$

where X_b denotes the global optimal solution. If the value of the moderator k is 1, then Eq. (13) is the

standard elite reverse learning. This kind of elite reverse learning has strict symmetry characteristics, with respect to the midpoint of the interval $(B^u + B^l)/2$ symmetry. The search can be strengthened in the local area, and the local optimal can be jumped out more effectively. It is less computationally complex. In this case, the formula can be expressed as

$$X_F = (B^u + B^l) - X_b \quad (14)$$

The specific steps of elite reverse learning are shown as follows.

Step 1 Find the global optimal solution X_b by comparing the fitness values.

Step 2 Assign the moderator k in Eq. (13) to 1.

Step 3 Calculate the inverse value of the elite X_F by Eq. (13).

3.3 Optimal retention mechanism

To enhance the convergence accuracy of the algorithm, an optimal value retention mechanism that builds upon the elite reverse learning strategy was introduced in this paper.

Initially, the elite reverse value is calculated using the elite reverse learning formula Eq. (13). This elite reverse value serves as an additional reference point alongside the current global optimal value during the update calculation for the individual positions within the population. Specifically, the midpoint between the position calculated from the optimal value and the position derived from the elite reverse value is computed. This intermediate position is then compared against the position obtained solely from the optimal value. The position yielding the better fitness value is selected as the final updated position for the individual.

By incorporating this mechanism, the algorithm can more effectively refine its search capabilities, improving both the precision and reliability of its convergence. This approach ensures a balance between exploring new potential solutions and exploiting known good solutions, thus enhancing overall performance.

3.4 Algorithm process

The algorithm flowchart is shown in Fig. 1.

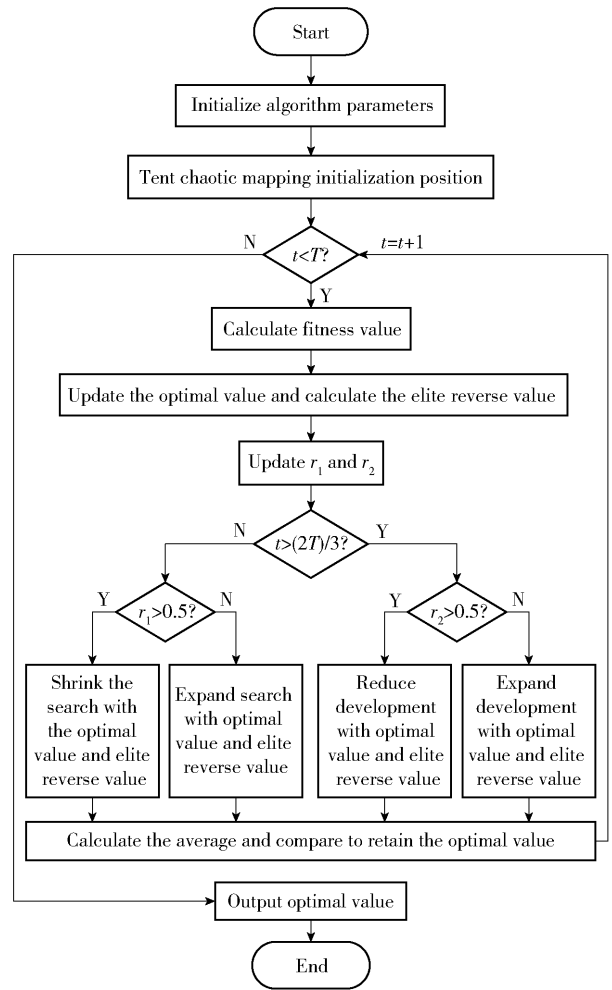


Fig. 1 Flowchart of RTEAO

The specific process of the RTEAO algorithm is shown as follows.

Step 1 Initialize the tent chaotic mapping population, and set the algorithm parameters.

Step 2 Judge whether the number of iterations reaches the maximum. If it reaches the maximum, end the iteration and output the optimal individual position. Otherwise, continue with Step 3.

Step 3 Calculate the fitness value of the individual position of the population and find the current global optimal individual.

Step 4 Get the reverse solution of the current optimal individual according to Eq. (13), and update the random numbers r_1 and r_2 , which are uniformly distributed in $(0,1)$.

Step 5 Simultaneously use the reverse solution and the global optimal solution obtained in Step 4 to update

the population position according to the search mechanism of the AO algorithm and take the middle position.

Step 6 Compare the middle position obtained in Step 5 with the position calculated by the optimal value, select the position with the better fitness value as the final update position, and return to Step 2.

3.5 Time complexity analysis

The time complexity of the RTEAO algorithm is an important factor in measuring the operation efficiency of the algorithm. It can be seen from the algorithm process that the time complexity of the RTEAO algorithm mainly consists of the following parts.

- 1) Initialize the population, the time complexity of which is $O(N)$.
- 2) Calculate the fitness value and the global optimum solution. The time complexity is $O(D)$, and the time complexity of elite reverse learning is $O(1)$.
- 3) The time complexity of population position update is $O(T)$, and the time complexity of taking the middle position and comparing the optimal value is $O(1)$.
- 4) Judge whether the algorithm iteration meets the termination condition, and the time complexity is $O(1)$.

Therefore, the time complexity of the RTEAO algorithm is $O(NTD)$, which is the same order of magnitude as the time complexity of the AO algorithm.

4 Experiment

4.1 Configuration environment

The experimental setup includes the following configurations. The central processing unit (CPU) is an Intel Core i7-12700H running at 2.30 GHz with 14 cores. The system’s random access memory (RAM) is 16 GB. The operating system is 64-bit Windows 11. And the development environment is Matlab 2020b. In order to evaluate the optimization performance of the RTEAO algorithm, a comparative experiment is conducted with several classical algorithms, including

the sine cosine algorithm (SCA), wind driven optimization (WDO), tuna swarm optimization (TSO), GWO, PSO, and the original AO algorithm. To ensure the reliability of the experimental outcomes, each algorithm is independently executed 30 times under identical conditions. Across all test scenarios, the maximum number of iterations is set to 500, with a population size of 50 individuals. The parameter settings for the algorithms are detailed in Table 1. It is noteworthy that the parameters for the RTEAO algorithm are kept consistent with those of the AO algorithm as outlined in Table 1.

Table 1 Algorithm parameter

Algorithm	Parameter	Value
PSO	Cognitive constant	2
	Social constant	2
	Minimum inertia weight	−2
	Maximum inertia weight	2
WDO	Pressure gradient term coefficient	3
	Gravitational constant	0.2
	Friction coefficient	0.4
	Coriolis force coefficient	0.4
GWO	Interval of convergence parameter	[2, 0]
SCA	Amplitude coefficient	2
TSO	Balance coefficient	0.7
	Mutation probability	0.05
AO	Step length ratio	0.1
	Intensity of the random disturbance term	0.1
	Rate of radius growth	0.026 5
	Angular velocity	0.005
RTEAO	Initial radius	10
	Step length ratio	0.1
	Intensity of the random disturbance term	0.1
	Rate of radius growth	0.026 5
RTEAO	Angular velocity	0.005
	Initial radius	10

4.2 Ablation experiment

RTEAO integrates tent chaotic mapping, elite reverse learning, and an optimal retention mechanism to enhance its performance. To validate the

effectiveness of these strategies, an ablation experiment using several comparison algorithms based on benchmark functions is conducted. The experimental setup includes TAO, TEAO, and RTEAO. TAO is an AO algorithm that solely employs the tent chaotic mapping strategy. TEAO is an AO algorithm that incorporates both the tent chaotic mapping and elite reverse learning strategies. RTEAO is an AO algorithm that utilizes all three strategies: tent chaotic mapping,

elite reverse learning, and the optimal retention mechanism.

Additionally, a comparative analysis against the original AO is performed to evaluate the improvements.

Table 2 presents the results of these ablation experiments, and Fig. 2 illustrates the convergence characteristics of the algorithms on selected benchmark functions, providing visual insights into their performance.

Table 2 Results of ablation experience on benchmark functions

Function	Algorithm	Mean value	Standard deviation	Optimal value	Function	Algorithm	Mean value	Standard deviation	Optimal value
F_1	AO	1.09×10^{-102}	5.95×10^{-102}	1.57×10^{-160}	F_{10}	AO	8.88×10^{-16}	0	8.88×10^{-16}
	TAO	1.72×10^{-106}	9.39×10^{-106}	1.59×10^{-160}		TAO	8.88×10^{-16}	0	8.88×10^{-16}
	TEAO	3.15×10^{-134}	1.60×10^{-133}	8.94×10^{-191}		TEAO	8.88×10^{-16}	0	8.88×10^{-16}
	RTEAO	4.27×10^{-182}	0	5.83×10^{-219}		RTEAO	8.88×10^{-16}	0	8.88×10^{-16}
F_2	AO	1.67×10^{-54}	8.74×10^{-54}	8.03×10^{-79}	F_{11}	AO	0	0	0
	TAO	2.21×10^{-56}	1.21×10^{-55}	1.05×10^{-82}		TAO	0	0	0
	TEAO	3.84×10^{-68}	1.44×10^{-67}	1.56×10^{-87}		TEAO	0	0	0
	RTEAO	2.06×10^{-92}	9.53×10^{-92}	1.62×10^{-104}		RTEAO	0	0	0
F_3	AO	4.86×10^{-107}	2.66×10^{-106}	1.09×10^{-157}	F_{12}	AO	7.90×10^{-6}	1.02×10^{106}	4.61×10^{-8}
	TAO	1.43×10^{-112}	7.83×10^{-112}	6.09×10^{-157}		TAO	7.69×10^{-6}	1.84×10^{106}	5.01×10^{-9}
	TEAO	4.70×10^{-130}	2.22×10^{-129}	5.53×10^{-184}		TEAO	4.25×10^{-6}	9.48×10^{106}	9.81×10^{-9}
	RTEAO	3.21×10^{-182}	0	6.02×10^{-206}		RTEAO	2.10×10^{-6}	3.66×10^{106}	4.17×10^{-10}
F_4	AO	2.55×10^{-59}	1.39×10^{-58}	2.82×10^{-80}	F_{13}	AO	2.24×10^{-5}	2.54×10^{106}	1.13×10^{-8}
	TAO	9.44×10^{-65}	5.17×10^{-64}	1.57×10^{-80}		TAO	2.96×10^{-5}	6.93×10^{106}	4.38×10^{-8}
	TEAO	1.92×10^{-67}	1.03×10^{-66}	3.18×10^{-93}		TEAO	1.50×10^{-5}	3.21×10^{106}	1.58×10^{-8}
	RTEAO	4.97×10^{-90}	2.47×10^{106}	1.53×10^{-108}		RTEAO	3.10×10^{-6}	8.68×10^{106}	2.19×10^{-9}
F_5	AO	2.10×10^{-3}	3.60×10^{-3}	1.06×10^{-5}	F_{15}	AO	4.69×10^{-4}	1.09×10^{106}	3.23×10^{-4}
	TAO	1.60×10^{-3}	2.30×10^{-3}	3.92×10^{-6}		TAO	5.00×10^{-4}	9.39×10^{106}	3.58×10^{-4}
	TEAO	6.50×10^{-3}	2.19×10^{-2}	3.61×10^{-7}		TEAO	5.76×10^{-4}	1.82×10^{106}	3.16×10^{-4}
	RTEAO	4.38×10^{-4}	8.74×10^{-4}	1.09×10^{-7}		RTEAO	4.10×10^{-4}	6.71×10^{106}	3.16×10^{-4}
F_6	AO	2.45×10^{-5}	3.56×10^{-5}	5.38×10^{-8}	F_{17}	AO	0.398 1	2.02×10^{-4}	0.397 9
	TAO	2.45×10^{-5}	3.72×10^{-5}	8.24×10^{-8}		TAO	0.398 2	4.10×10^{-4}	0.397 9
	TEAO	2.44×10^{-5}	5.55×10^{-5}	2.42×10^{-9}		TEAO	0.412 2	1.66×10^{-2}	0.397 9
	RTEAO	7.97×10^{-6}	2.27×10^{-5}	2.74×10^{-11}		RTEAO	0.398 1	2.64×10^{-4}	0.397 9
F_7	AO	1.55×10^{-4}	1.59×10^{-4}	1.77×10^{-6}	F_{18}	AO	3.030 4	0.042 7	3.100 0
	TAO	1.05×10^{-4}	9.31×10^{-5}	3.42×10^{-6}		TAO	3.032 4	0.049 2	3.120 0
	TEAO	1.01×10^{-4}	9.03×10^{-5}	8.95×10^{-7}		TEAO	3.861 6	0.807 0	3.200 0
	RTEAO	1.28×10^{-4}	1.27×10^{-4}	1.31×10^{-6}		RTEAO	3.028 3	0.027 0	3.120 0
F_8	AO	-3.51×10^3	8.67×10^2	-4.19×10^3					
	TAO	-3.52×10^3	9.06×10^2	-4.19×10^3					
	TEAO	-1.93×10^3	3.18×10^2	-2.99×10^3					
	RTEAO	-3.65×10^3	8.44×10^2	-4.18×10^3					

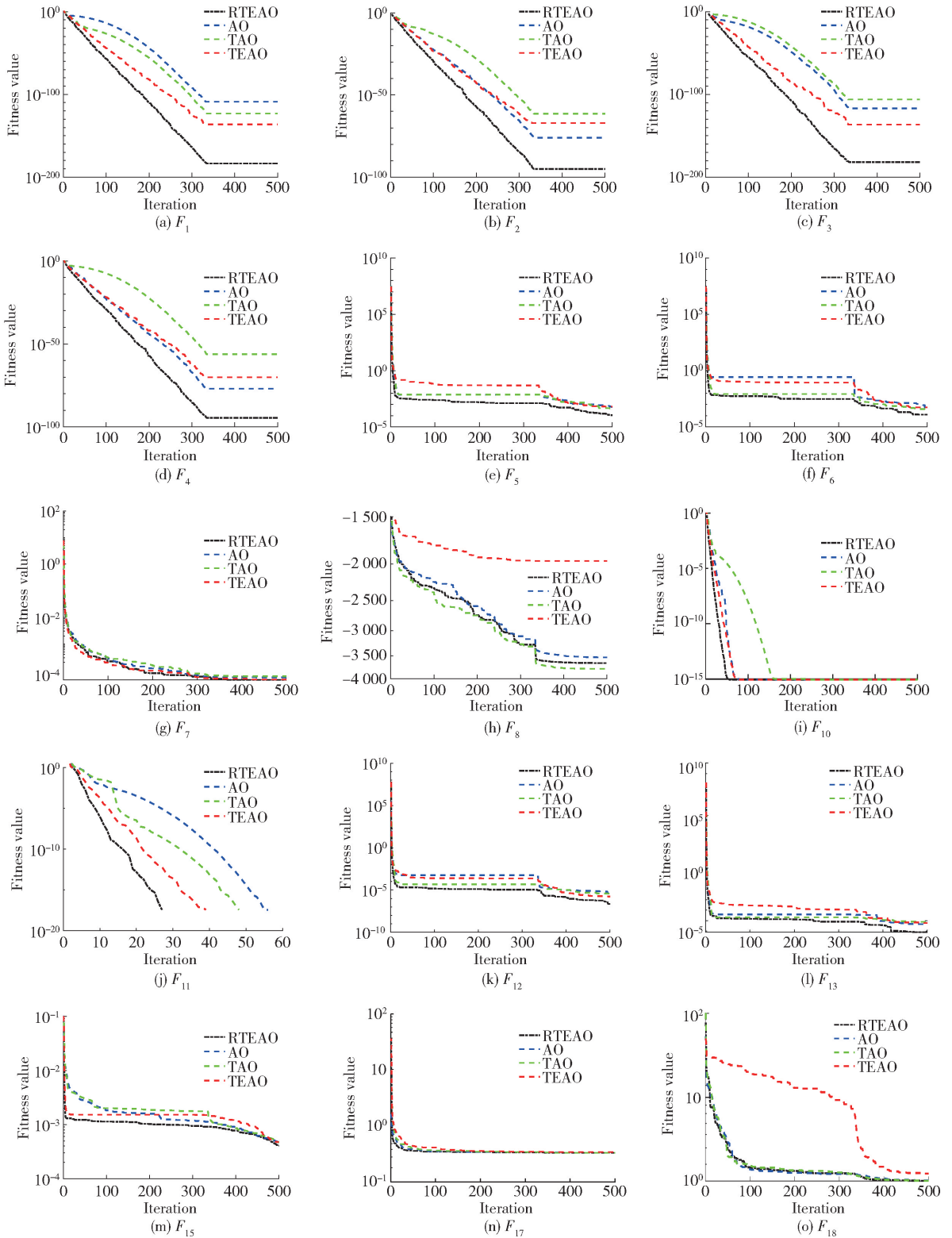


Fig. 2 Mean convergence curve of different algorithms on benchmark functions

From Table 2, it is evident that the RTEAO algorithm leads to a decrease in the optimal value,

mean value, and standard deviation for functions F_1 to F_5 . This indicates that the newly incorporated

strategies significantly enhance the algorithm’s performance.

Specifically, for TAO, the optimal and mean values for functions F_6 , F_7 , and F_8 are similar to those of the original AO algorithm. This suggests that solely relying on tent chaotic mapping does not intuitively improve the search diversity or overall performance of the algorithm.

For TEAO, it shows improved test results on functions F_{15} , F_{17} , and F_{18} compared to both AO and TAO. This highlights that the combination of these two strategies effectively boosts the optimization performance of the algorithm.

For RTEAO, it performs exceptionally well across all tested functions. Its mean value and standard deviation remain at low levels relative to TAO and TEAO, demonstrating further improvements in stability and optimization accuracy. The comprehensive incorporation of these strategies enhances both the reliability and precision of the algorithm.

Additionally, Fig. 2 provides convergence curves for various unimodal and multimodal functions,

demonstrating that the RTEAO algorithm achieves high convergence accuracy and demonstrates rapid convergence capabilities when handling these functions. These results confirm that RTEAO not only improves upon the baseline performance metrics but also accelerates the convergence process, making it a robust choice for optimization tasks.

4.3 Comparative experiment

To objectively assess the efficacy of the RTEAO algorithm in solving complex optimization problems, its performance is benchmarked against the aforementioned classical algorithms across a suite of benchmark functions.

The experimental results, presented in Table 3, show a comprehensive overview of how RTEAO stacks up against these established algorithms. Table 3 includes metrics such as the optimal value, mean value, and standard deviation across various benchmark functions, allowing for a detailed comparison of solution quality and consistency.

Table 3 Experiment results of the comparison algorithm on selected benchmark functions

Function	Algorithm	Mean value	Standard deviation	Optimal value	Function	Algorithm	Mean value	Standard deviation	Optimal value
F_1	GWO	2.51×10^{-33}	2.63×10^{-33}	7.09×10^{-35}	F_4	GWO	2.69×10^{-8}	5.14×10^{-8}	3.68×10^{-9}
	SCA	6.09	12.93	1.50×10^{-2}		SCA	29.02	12.35	9.57
	PSO	1.44×10^{-1}	1.99	9.41		PSO	1.57	0.13	1.28
	TSO	2.40×10^{-240}	0	2.09×10^{-272}		TSO	2.27×10^{-118}	7.28×10^{-118}	4.67×10^{-129}
	WDO	6.47×10^{-18}	3.07×10^{-17}	4.82×10^{-23}		WDO	3.26×10^{-9}	4.82×10^{-9}	3.19×10^{-12}
	AO	2.67×10^{-134}	1.46×10^{-133}	3.01×10^{-163}		AO	2.77×10^{-76}	1.47×10^{-75}	4.15×10^{-84}
	RTEAO	4.65×10^{-186}	0	8.33×10^{-219}		RTEAO	1.16×10^{-94}	5.14×10^{-94}	7.44×10^{-118}
F_2	GWO	1.39×10^{-18}	1.22×10^{-18}	1.41×10^{-19}	F_5	GWO	26.93	0.86	25.42
	SCA	3.94×10^{-2}	4.43×10^{-2}	1.80×10^{-3}		SCA	5.69×10^5	1.00×10^6	2.41×10^3
	PSO	4.99×10	2.73×10^{-33}	15.76		PSO	3.72×10^3	1.38×10^3	1.92×10^3
	TSO	1.73×10^{-117}	9.49×10^{-117}	2.30×10^{-135}		TSO	3.29	8.51	1.40×10^{-6}
	WDO	1.03×10^{-8}	1.48×10^{-8}	1.14×10^{-10}		WDO	28.65	3.04×10^{-2}	28.59
	AO	5.06×10^{-55}	2.77×10^{-54}	3.86×10^{-83}		AO	1.09×10^{-2}	2.48×10^{-2}	8.68×10^{-5}
	RTEAO	4.87×10^{-94}	1.46×10^{-93}	2.89×10^{-106}		RTEAO	5.89×10^{-4}	1.00×10^{-3}	6.40×10^{-7}
F_3	GWO	1.17×10^{-8}	3.67×10^{-8}	5.79×10^{-11}	F_6	GWO	0.43	0.34	2.96×10^{-5}
	SCA	7.01×10^3	4.36×10^3	4.79×10^{-2}		SCA	8.68	9.35	3.58
	PSO	68.04	14.66	40.03		PSO	14.63	1.94	9.19
	TSO	2.82×10^{-216}	0	1.06×10^{-262}		TSO	1.07×10^{-5}	1.69×10^{-5}	1.27×10^{-7}
	WDO	5.52×10^{-14}	1.37×10^{-13}	4.16×10^{-17}		WDO	7.43×10^{-2}	3.16×10^{-2}	3.08×10^{-2}
	AO	4.01×10^{-106}	2.20×10^{-105}	3.89×10^{-163}		AO	1.48×10^{-4}	5.35×10^{-4}	1.16×10^{-7}
	RTEAO	9.36×10^{-187}	0	3.72×10^{-222}		RTEAO	1.20×10^{-5}	4.57×10^{-5}	5.31×10^{-9}

Table 3Continued

Function	Algorithm	Mean value	Standard deviation	Optimal value	Function	Algorithm	Mean value	Standard deviation	Optimal value
F_7	GWO	1.40×10^{-3}	7.18×10^{-4}	4.87×10^{-4}	F_{13}	GWO	4.19×10^{-1}	0.21	0.10
	SCA	9.98×10^4	2.25×10^5	3.23		SCA	1.01×10^6	3.39×10^6	5.02
	PSO	2.63×10^2	62.83	1.28×10^2		PSO	2.19	0.25	1.63
	TSO	2.52×10^{-4}	1.91×10^{-4}	1.53×10^{-5}		TSO	3.20×10^{-3}	9.90×10^{-3}	1.49×10^{-8}
	WDO	2.14×10^{-5}	1.40×10^{-4}	9.70×10^{-6}		WDO	3.06×10^{-1}	0.64	2.40×10^{-2}
	AO	5.17×10^{-5}	5.22×10^{-5}	1.88×10^{-6}		AO	2.64×10^{-5}	4.40×10^{-5}	3.47×10^{-8}
	RTEAO	6.71×10^{-5}	6.34×10^{-5}	1.91×10^{-6}		RTEAO	9.48×10^{-7}	1.53×10^{-6}	8.71×10^{-9}
F_8	GWO	-1.22×10^3	1.00×10^2	-1.42×10^3	F_{15}	GWO	1.10×10^{-3}	4.21×10^{-4}	8.73×10^{-4}
	SCA	-9.26×10^3	77.69	-1.12×10^3		SCA	1.00×10^{-3}	2.64×10^{-5}	1.00×10^{-3}
	PSO	-1.23×10^3	1.19×10^2	-1.47×10^3		PSO	1.10×10^{-3}	3.65×10^{-5}	9.91×10^{-4}
	TSO	-1.91×10^3	5.04×10^{-6}	-1.91×10^3		TSO	1.00×10^{-3}	2.88×10^{-5}	1.00×10^{-3}
	WDO	-1.10×10^3	87.97	-1.38×10^3		WDO	1.20×10^{-3}	5.24×10^{-4}	3.08×10^{-4}
	AO	-1.59×10^3	3.31×10^2	-1.90×10^3		AO	1.00×10^{-3}	5.75×10^{-4}	3.52×10^{-4}
	RTEAO	-1.16×10^3	3.75×10^2	-1.89×10^3		RTEAO	1.20×10^{-3}	7.40×10^{-4}	3.91×10^{-4}
F_{10}	GWO	20.88	7.74×10^{-2}	20.69	F_{17}	GWO	3.99×10^{-1}	7.20×10^{-3}	3.97×10^{-1}
	SCA	20.29	6.21×10^{-2}	20.17		SCA	4.09×10^{-1}	1.80×10^{-2}	3.98×10^{-1}
	PSO	18.15	6.32	4.13		PSO	3.98×10^{-1}	1.90×10^{-4}	3.97×10^{-1}
	TSO	8.88×10^{-16}	0	8.88×10^{-16}		TSO	3.98×10^{-1}	3.24×10^{-16}	3.97×10^{-1}
	WDO	0.69	3.82	1.54×10^{-10}		WDO	4.31×10^{-1}	1.83×10^{-6}	3.97×10^{-1}
	AO	8.88×10^{-16}	0	8.88×10^{-16}		AO	3.98×10^{-1}	2.13×10^{-4}	3.97×10^{-1}
	RTEAO	8.88×10^{-16}	0	8.88×10^{-16}		RTEAO	4.04×10^{-1}	7.30×10^{-3}	3.99×10^{-1}
F_{11}	GWO	1.20×10^{-3}	3.20×10^{-3}	0	F_{18}	GWO	8.40	20.55	3.00
	SCA	0.43	0.23	2.53×10^{-2}		SCA	3.00	4.53×10^{-5}	3.00
	PSO	0.54	5.26×10^{-2}	0.43		PSO	3.01	1.43×10^{-2}	3.00
	TSO	0	0	0		TSO	2.99	3.12×10^{-15}	2.99
	WDO	2.69×10^{-3}	1.08×10^{-2}	0		WDO	3.00	7.49×10^{-4}	3.00
	AO	0	0	0		AO	3.02	2.72×10^{-2}	3.00
	RTEAO	0	0	0		RTEAO	3.59	0.90	3.00
F_{12}	GWO	2.82×10^{-2}	1.74×10^{-2}	5.90×10^{-3}					
	SCA	1.51×10^5	3.01×10^5	1.05					
	PSO	3.35	3.23	0.43					
	TSO	5.87×10^{-7}	8.45×10^{-7}	5.19×10^{-12}					
	WDO	5.78×10^{-2}	0.13	1.80×10^{-3}					
	AO	3.35×10^{-6}	6.51×10^{-6}	3.47×10^{-8}					
	RTEAO	1.65×10^{-7}	2.64×10^{-7}	2.95×10^{-11}					

Fig. 3 illustrates the convergence curves of these algorithms on selected benchmark functions, offering visual insights into their convergence behaviors. Fig. 3 helps to demonstrate not only the final solution quality but also the rate at which each algorithm converges to

its optimal solution. From these results, it can be observed that the RTEAO algorithm achieves higher convergence accuracy more rapidly compared to the other algorithms.

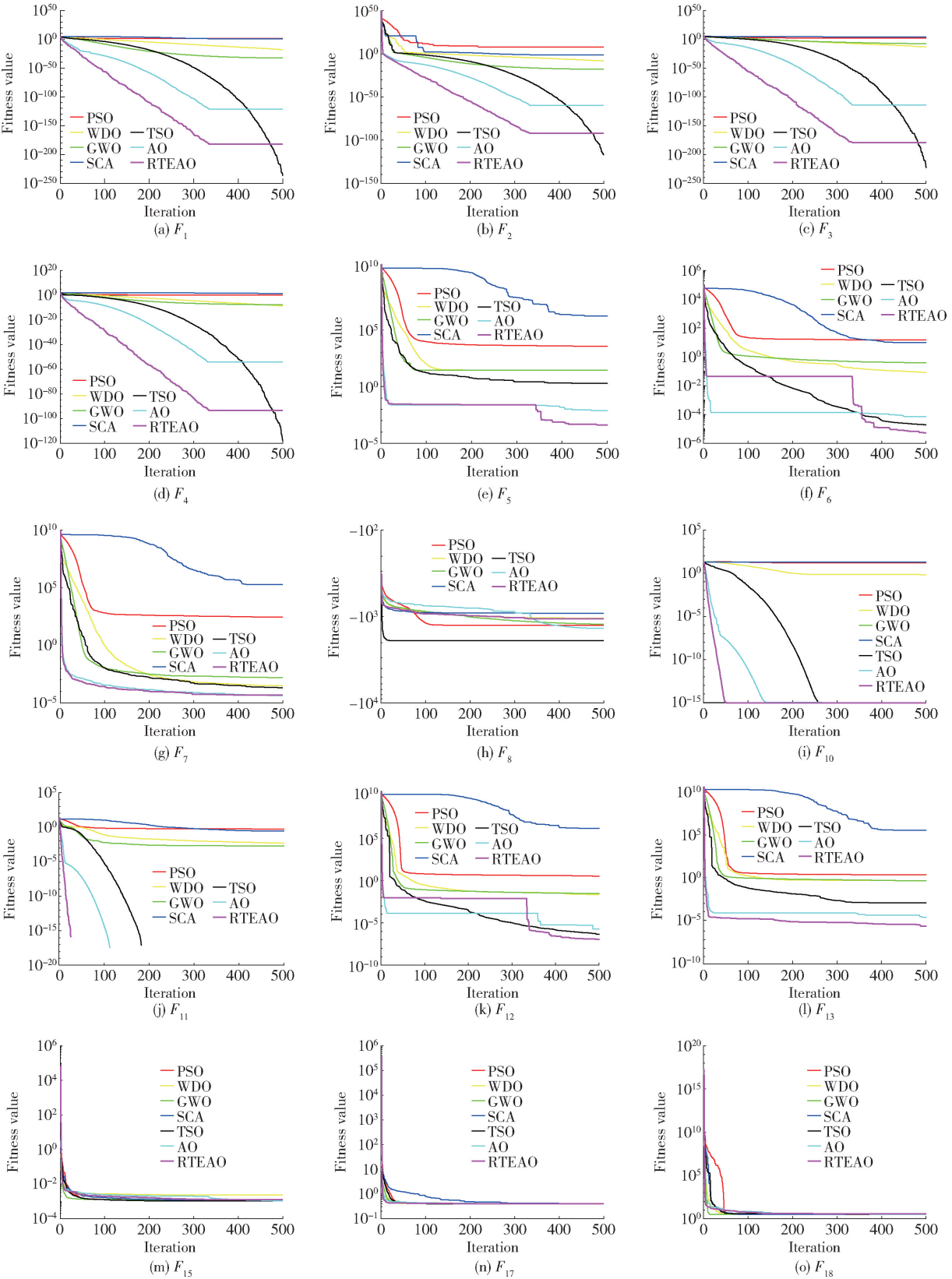


Fig. 3 Convergence curve of different algorithms in the comparison experiment

4.3.1 Experimental results and analysis

From Table 3, it is evident that the RTEAO algorithm demonstrates strong performance when handling most unimodal functions, $F_1 - F_6$. It not only efficiently locates the optimal solution but also excels the mean value and standard deviation. This highlights the RTEAO’s robust global search capability and stable optimization performance for unimodal functions.

For multimodal functions, such as F_{12} and F_{13} , although the optimization performance of RTEAO algorithm is slightly lower compared to its handling of unimodal functions, it still outperforms the other comparative algorithms. When dealing with multimodal functions, such as F_{15} , F_{17} , and F_{18} , while the mean value and standard deviation may not be the best among all algorithms, they remain at a relatively small order of magnitude. This can be attributed to RTEAO’s ability to comprehensively evaluate multiple candidate solutions, enabling the algorithm to perform localized, in-depth searches between existing candidate positions. RTEAO algorithm not only enhances the capacity to explore the global optimal solution but also ensures good stability and high search efficiency.

Fig. 3 further reinforces these findings by illustrating that RTEAO algorithm achieves higher convergence precision and faster convergence speed compared to the other algorithms. These results underscore the RTEAO’s strong competitiveness within the realm of swarm intelligence optimization techniques.

4.3.2 Statistical test

In order to evaluate the reliability of the RTEAO algorithm and exclude accidental differences, the Wilcoxon rank sum test is used to evaluate the significant difference between RTEAO and the comparison algorithms. If the significance level is less than the selected significance level of 0.05, reject the null hypothesis, indicating that there is a significant difference between the two samples. Otherwise, there is no significant difference. Table 4 shows the significance level of RTEAO and the comparison algorithms. In Table 4, rank-biserial correlation is the significance judgment result, and the symbols “+”, “-”, and “=” indicate that the RTEAO algorithm is superior to, inferior to, and equal to other algorithms, respectively.

Table 4 Experimental results of the Wilcoxon rank sum test

Function	Probability obtained from Wilcoxon rank sum test						Rank-biserial correlation					
	PSO	WDO	TSO	GWO	SCA	AO	PSO	WDO	TSO	GWO	SCA	AO
F_1	0.028 6	0.028 6	0.885 7	0.028 6	0.028 6	0.028 6	+	+	-	+	+	+
F_2	0.028 6	0.028 6	0.485 7	0.028 6	0.028 6	0.028 6	+	+	=	+	+	+
F_3	0.028 6	0.028 6	0.142 9	0.028 6	0.028 6	0.028 6	+	+	=	+	+	+
F_4	0.028 6	0.028 6	0.028 6	0.028 6	0.028 6	0.028 6	+	+	+	+	+	+
F_5	0.028 6	0.028 6	0.028 6	0.028 6	0.028 6	0.200 0	+	+	+	+	+	=
F_6	0.028 6	0.028 6	0.028 6	0.485 7	0.028 6	0.028 6	+	+	+	=	+	+
F_7	0.028 6	0.200 0	0.028 6	0.028 6	0.028 6	0.685 7	+	=	+	+	+	=
F_8	0.885 7	0.885 7	0.028 6	0.885 7	0.142 9	0.685 7	=	=	+	=	=	=
F_9	0.028 6	0.028 6	0.028 6	0.142 9	0.028 6	NaN	+	+	+	=	+	=
F_{10}	0.028 6	0.028 6	NaN	0.028 6	0.028 6	NaN	+	+	=	+	+	=
F_{11}	0.028 6	0.142 9	NaN	0.142 9	0.142 9	NaN	+	=	=	=	=	=
F_{12}	0.028 6	0.028 6	0.028 6	0.028 6	0.028 6	0.028 6	+	+	+	+	+	+
F_{13}	0.028 6	0.028 6	0.028 6	0.142 9	0.142 9	0.028 6	+	+	+	=	+	+
F_{15}	0.028 6	0.885 7	0.142 9	0.028 6	0.885 7	0.885 7	+	=	=	+	=	=
F_{17}	0.028 6	0.485 7	0.028 6	0.685 7	0.685 7	0.485 7	+	=	+	=	=	=
F_{18}	0.485 7	0.200 0	0.485 7	0.200 0	0.200 0	0.485 7	=	=	=	=	=	=

It can be seen from Table 4 that the rank-biserial correlation on the unimodal function $F_1 - F_6$ is “+”, which indicates that the RTEAO algorithm is significantly different from the comparison algorithms, further confirming its reliability and strong development capability. On the multimodal functions F_9, F_{10} , and F_{11} , the comparison results of RTEAO and AO algorithms show that the rank-biserial correlation is “=”, which means that the performance of the two functions is equivalent, and there is no significant difference between them. On multimodal functions F_{12}, F_{13} , and F_{15} , although RTEAO algorithm is slightly better than AO algorithm in terms of the mean value, standard deviation, and optimal value, the rank-biserial correlation is “+” compared with other comparison algorithms, which indicates that RTEAO algorithm still shows a strong competitive advantage even on these more challenging test functions. To sum up, the RTEAO algorithm shows strong reliability and stability on the whole.

4.4 Algorithm application

4.4.1 Theoretical underpinnings of algorithm application

To improve the performance of the BP neural network, the RTEAO algorithm is employed to optimize its critical parameters, particularly the initial weights and biases, as well as the adaptive adjustment of hyperparameters throughout the training process.

1) Initialization of weight and bias

Rather than initializing the weights and biases of the BP neural network with random or predefined values, the RTEAO algorithm is employed to determine their initial configuration. By harnessing the strong exploration capabilities of RTEAO algorithm, this approach seeks a more optimal initialization point, which can facilitate faster convergence and yield improved overall network performance.

2) Parameter adjustment during training

Throughout the training process, RTEAO algorithm continues to refine the weights and biases of the BP neural network. By dynamically adjusting these parameters based on the elite reverse learning strategy, tent chaotic mapping, and the optimal retention

mechanism, the algorithm seeks to avoid local optima and improve the robustness and accuracy of the BP neural network.

4.4.2 Evaluation of the improved BP neural network with RTEAO algorithm

A comparative study is conducted to evaluate the effectiveness of the improved BP neural network using two widely used benchmark datasets: the Boston Housing Price Dataset for regression tasks and the Wisconsin Breast Cancer Dataset for classification tasks.

The first step involves defining a fitness function that evaluates the performance of the BP neural network. This function serves as a guide for the RTEAO algorithm to search for the optimal solution within the parameter space. During each iteration of the RTEAO algorithm, the search space is explored by adjusting the positions of individuals. The RTEAO algorithm effectively combines local and global search mechanisms. Local search focuses on refining current solutions to improve their quality. Global search ensures exploration of the entire search space to avoid getting trapped in local optima. These mechanisms work together to efficiently approach the optimal solution.

1) Convergence towards the global optimum

As the number of iterations increases, the RTEAO algorithm progressively converges towards the global optima. It leverages strategies such as tent chaotic mapping, elite reverse learning, and optima retention mechanism to maintain diversity and enhance convergence speed and accuracy.

2) Output optimal parameter

Upon completion of all iterations, the RTEAO algorithm outputs a set of optimized weights and biases. These parameters represent the best solution found within the search space.

3) Initialize the parameters of the BP neural network

The optimal weights and biases obtained from the RTEAO algorithm are then used to initialize the BP neural network. This initialization provides a more favorable weight and bias for training, potentially leading to faster convergence and improved performance.

4) Training the optimized BP neural network with

RTEAO algorithm

Finally, the BP neural network initialized with these optimized parameters undergoes standard training. During this phase, the network further adjusts its weights and biases using the BP algorithm until it meets the preset target error threshold or reaches the maximum number of iterations.

This comprehensive approach ensures that the BP neural network benefits from both the global search capabilities of the RTEAO algorithm and the refinement abilities of the BP training process, ultimately enhancing its predictive accuracy and robustness. Fig. 4 is a detailed flowchart of the algorithm application process. To evaluate and compare the effects of the RTEAO and AO algorithms on improving the performance of a BP neural network, this study conducted experiments under identical conditions. Each dataset is partitioned into separate training and test sets. Specifically, for both the Boston Housing Price Dataset and Wisconsin Breast Cancer Dataset, samples are randomly split according to a fixed ratio. The following evaluation metrics are calculated for the original BP, RTEAO optimized BP, and AO optimized BP on both the training set and the test set.

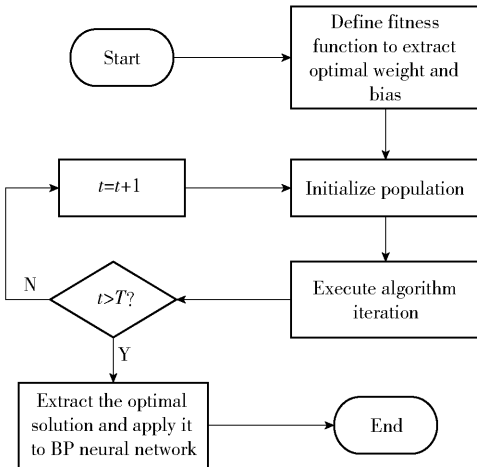


Fig. 4 Flowchart of the optimized BP neural network with RTEAO algorithm

4.4.3 Experimental validation and analysis

Root mean square error (RMSE) is used to measure the mean magnitude of prediction errors, penalizing larger deviations more heavily. Mean absolute error (MAE) represents the mean absolute difference between predicted and actual values, providing a

straightforward measure of error. Coefficient of determination R^2 indicates how well the model explains the variance in the data, with higher values reflecting superior performance.

For visual comparison, bar charts are also presented. As shown in Figs. 5 – 6, the experimental results based on the Boston Housing Dataset and the Wisconsin Breast Cancer Dataset are presented, respectively.

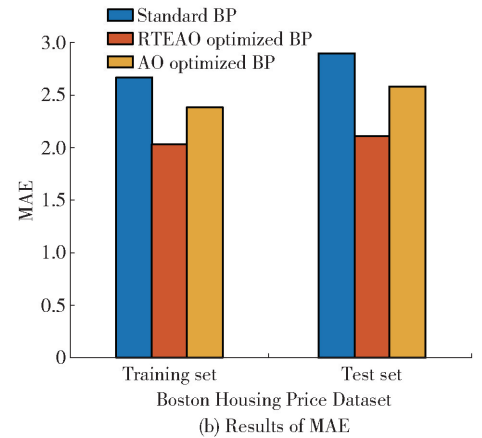


Fig. 5 Predicted results for the Boston Housing Price Dataset

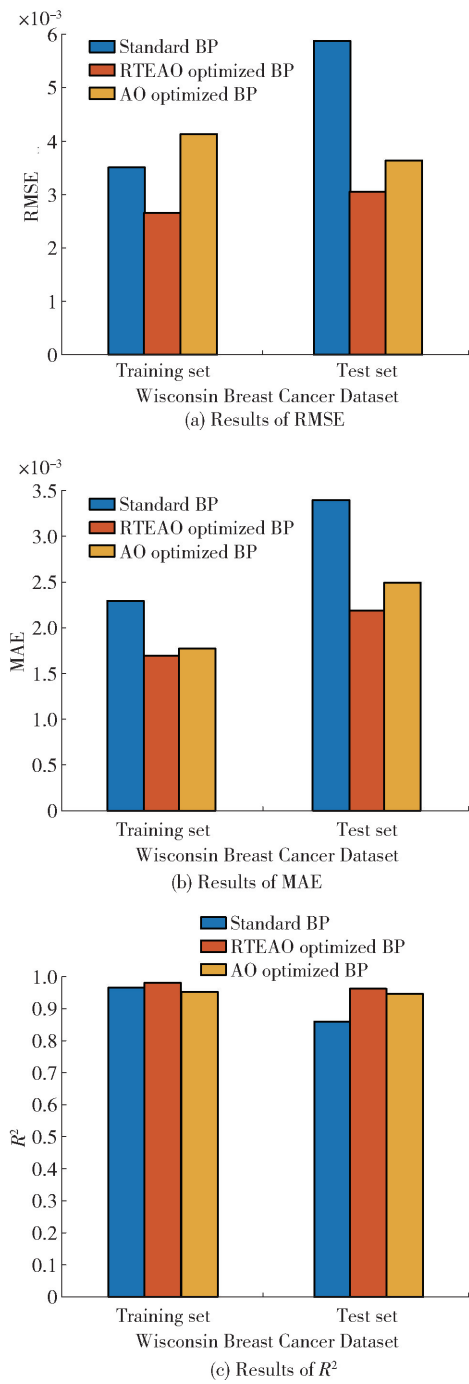


Fig.6 Predicted results for the Wisconsin Breast Cancer Dataset

In Figs.5 – 6, standard BP refers to the original BP neural network. RTEAO optimized BP denotes a BP neural network that was optimized using the RTEAO algorithm. AO optimized BP indicates a BP neural network that was optimized with the AO algorithm.

The experimental results indicate that both the

RTEAO and AO algorithms significantly enhance the performance of the BP neural network, but the RTEAO optimized BP neural network demonstrates superior outcomes in several key metrics. The RTEAO optimized BP neural network achieves lower RMSE values than the AO optimized BP neural network. This suggests that the predictions made by the RTEAO optimized BP neural network are closer to the actual values, indicating a higher level of accuracy.

Similarly, the RTEAO optimized BP neural network also shows reduced MAE values. Lower MAE signifies that the absolute differences between predicted and actual values are smaller, reinforcing the improved accuracy of the RTEAO optimized BP neural network.

The RTEAO optimized BP neural network exhibits advantages in terms of R^2 values as well. A higher R^2 score indicates that the RTEAO optimized BP neural network explains a greater proportion of the variance in the data, suggesting better generalization capabilities and predictive power for unseen data.

These findings highlight the strong applicability and effectiveness of the RTEAO algorithm in optimizing BP neural networks. The ability of the RTEAO algorithm to improve the generalization ability and prediction accuracy makes it a valuable tool, especially for those requiring high precision and reliability. The experimental evidence supports the conclusion that the RTEAO algorithm is capable of delivering optimal prediction performance, making it a preferred choice for enhancing the performance of the BP neural network.

5 Conclusions

In this paper, RTEAO algorithm was proposed. Firstly, a tent chaotic mapping is introduced in the population initialization phase to enhance the diversity and distribution uniformity of initial solutions. Secondly, the elite reverse learning strategy is used to calculate the elite reverse value when updating the global optimal value, and combine it with the current optimal value to apply to the iterative update of individual positions, effectively preventing the algorithm from falling into the local optimal solution

prematurely, and expanding the search range. Finally, the optimal reservation mechanism is introduced to select individuals with better fitness values as the final update location in the population location update phase. The above strategies are designed to enable the RTEAO algorithm to continuously approach the global optimal solution during the iterative process. Experimental results show that the convergence speed and convergence accuracy of the RTEAO algorithm on the CEC2005 benchmark function set are better than those of the comparison algorithm. Furthermore, the RTEAO algorithm achieved excellent results in the parameters of BP optimization experiments.

References

- [1] EFFATPARVAR M, GARSHASBI M S. A genetic algorithm for static load balancing in parallel heterogeneous systems. *Procedia: Social and Behavioral Sciences*, 2014, 129: 358 – 364.
- [2] LYU W C, GUO X Z, SHEN Y Z, et al. A genetic algorithm-calibrated heat source model for reconstructing thermal history and precipitate evolution in additive friction stir multilayer deposition. *CIRP Journal of Manufacturing Science and Technology*, 2025, 61: 70 – 87.
- [3] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer. *Advances in Engineering Software*, 2014, 69: 46 – 61.
- [4] MIRJALILI S, LEWIS A. The whale optimization algorithm. *Advances in Engineering Software*, 2016, 95: 51 – 67.
- [5] DORIGO M, BIRATTARI M, STÜTZLE T. Ant colony optimization: artificial ants as a computational intelligence technique. *IEEE Computational Intelligence Magazine*, 2006, 1(4): 28 – 39.
- [6] EL-SHAMIR EZUGWU A, ADEWUMI A O, FRÎNCU M E. Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem. *Expert Systems with Applications*, 2017, 77: 189 – 210.
- [7] ABUALIGAH L, YOUSRI D, ABD ELAZIZ M, et al. Aquila optimizer: a novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, 2021, 157: Article 107250.
- [8] WANG S, JIA H M, LIU Q X, et al. An improved hybrid Aquila optimizer and Harris Hawks optimization for global optimization. *Mathematical Biosciences and Engineering*, 2021, 18(6): 7076 – 7109.
- [9] HEIDARI A A, MIRJALILI S, FARIS H, et al. Harris Hawks optimization: algorithm and applications. *Future Generation Computer Systems*, 2019, 97: 849 – 872.
- [10] FATANI A, DAHOU A, AL-QANESS M A, et al. Advanced feature extraction and selection approach using deep learning and Aquila optimizer for IoT intrusion detection system. *Sensors*, 2022, 22(1): Article 140.
- [11] ZHANG Y J, YAN Y X, ZHAO J, et al. AOAAO: the hybrid algorithm of arithmetic optimization algorithm with Aquila optimizer. *IEEE Access*, 2022, 10: 10907 – 10933.
- [12] ABUALIGAH L, DIABAT A, MIRJALILI S, et al. The arithmetic optimization algorithm. *Computer Methods in Applied Mechanics and Engineering*, 2021, 376: Article 113609.
- [13] EL MARZOUGUI A, BAHSENE S, OUKENNOU A, et al. Enhanced algorithm for hybrid renewable energy systems, optimized with battery storage; a case study in Dakhla region, Morocco. *Journal of Energy Storage*, 2025, 120: Article 116386.
- [14] WU S Q, HE B T, ZHANG J, et al. PSAO: an enhanced Aquila optimizer with particle swarm mechanism for engineering design and UAV path planning problems. *Alexandria Engineering Journal*, 2024, 106: 474 – 504.
- [15] ZHANG Y, XU X P, ZHANG N, et al. Adaptive Aquila optimizer combining niche thought with dispersed chaotic swarm. *Sensors*, 2023, 23(2): Article 755.
- [16] ZHANG K P, LIU Y H, WANG X, et al. IBMRFO: improved binary manta ray foraging optimization with chaotic tent map and adaptive somersault factor for feature selection. *Expert Systems with Applications*, 2024, 251: Article 123977.
- [17] QI Y Y, JIANG A P, GAO Y H. A Gaussian convolutional optimization algorithm with tent chaotic mapping. *Scientific Reports*, 2024, 14(1): Article 31027.
- [18] SINGH N J, SINGH S, CHOPRA V, et al. Chaotic evolutionary programming for an engineering optimization problem. *Applied Sciences-Basel*, 2021, 11(6): Article 2717.
- [19] JIAO C Y, YU K J, ZHOU Q L. An opposition-based learning adaptive chaotic particle swarm optimization algorithm. *Journal of Bionic Engineering*, 2024, 21(6): 3076 – 3097.
- [20] LIU C, LIU Y Q, WU T, et al. An optimization algorithm for solving high-dimensional complex functions based on a multipopulation cooperative bare-bones particle swarm. *Journal of Electrical Engineering & Technology*, 2022, 17(4): 2441 – 2456.
- [21] ZHANG H Z, ZHANG Y, NIU Y X, et al. A grey wolf optimizer combined with artificial fish swarm algorithm for engineering design problems. *Ain Shams Engineering Journal*, 2024, 15(7): Article 102797.
- [22] LIU J S, HOU Y L, LI Y, et al. Advanced strategies on update mechanism of tree-seed algorithm for function optimization and engineering design problems. *Expert Systems with Applications*, 2024, 236: Article 121312.
- [23] JOVANOVIĆ M, RAKIĆ T, JANCIC-STOJANOVIĆ B, et al. Retention mechanism assessment and method development for the analysis of iohexol and its related compounds in hydrophilic interaction liquid chromatography. *Analytical and Bioanalytical Chemistry*, 2014, 406(17): 4217 – 4232.
- [24] XU H P, TANG Y L, MAO W J, et al. Exploration of the optimal retention method in vivo for stem cell therapy: low-intensity ultrasound preconditioning. *Regenerative Therapy*, 2025, 29: 484 – 492.

(Editor: Wang Xuying)