Journal of Computer and Communications, 2024, 12, 151-170
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Arctic Puffin Optimization Algorithm Based on Multi-Strategy Blending

Ling Sun, Bo Wang*

College of Science, Shenyang University of Technology, Shenyang, China
Email: 18841694661@163.com, *5692188@qq.com

How to cite this paper: Sun, L. and Wang, B. (2024) Arctic Puffin Optimization Algorithm Based on Multi-Strategy Blending. *Journal of Computer and Communications*, **12**, 151-170.
https://doi.org/10.4236/jcc.2024.1212010

Received: November 21, 2024
Accepted: December 27, 2024
Published: December 30, 2024

Abstract

A hybrid strategy is proposed to solve the problems of poor population diversity, insufficient convergence accuracy and susceptibility to local optimal values in the original Arctic Puffin Optimization (APO) algorithm, Enhanced Tangent Flight Adaptive Arctic Puffin Optimization with Elite initialization and Adaptive *t*-distribution Mutation (ETAAPO). Elite initialization improves initial population quality and accelerates convergence. Tangent Flight of the Tangent search algorithm replaces Levy Flight to balance local search and global exploration. The adaptive *t*-distribution mutation strategy enhances the optimization ability. ETAAPO was tested on CEC2021 functions, Wilcoxon rank-sum tests, and engineering problems, demonstrating superior optimization performance and faster convergence.

Keywords

Arctic Puffin Optimization, Elite Reverse Learning Strategy, Tangential Flight Strategy, Adaptive *t*-Distribution Variation Strategy

1. Introduction

Optimization algorithms, as a key interdisciplinary technology, have demonstrated extensive application value and profound impact across various fields, both domestically and internationally. In China, their applications have penetrated into key areas such as intelligent transportation [1], financial investment [2], intelligent manufacturing [3], and energy management [4], significantly enhancing system efficiency, reducing costs, and promoting green and sustainable development. On the global stage, optimization algorithms play an indispensable role in cutting-edge fields such as aerospace, bioinformatics, e-commerce, and environmental protection, driving technological progress and societal development. Therefore,

mastering and delving into the study of optimization algorithms is of great significance for advancing scientific and technological innovation and fostering socio-economic development.

In recent years, with the remarkable improvement in computer performance and the rapid evolution of electronic information technology, the field of optimization algorithms has ushered in opportunities for vigorous growth. Numerous innovative optimization algorithms have emerged, including but not limited to genetic algorithms [5], particle swarm optimization [6], firefly optimization algorithm [7], salp swarm optimization algorithm [8], artificial bee colony optimization algorithm [9], butterfly optimization algorithm [10], and sparrow search algorithm [11]. These algorithms have achieved significant research outcomes within their respective domains and have demonstrated powerful performance in practical applications. With the continuous advancement of computational capabilities and algorithmic theory, these algorithms are expected to be applied and developed in more fields in the future.

However, faced with increasingly complex and large-scale optimization problems, original optimization algorithms have gradually revealed limitations such as insufficient convergence accuracy, slow convergence speed, and susceptibility to local optima, which fail to meet the efficient optimization demands in practical applications. To overcome these challenges, domestic researchers have actively explored and implemented various improvement strategies. For instance, Ma *et al.* [12] optimized the key parameters A and the position update mechanism in the whale optimization algorithm by introducing a nonlinear convergence factor and adaptive inertia weight, effectively balancing the algorithm's global search and local exploration capabilities and significantly enhancing convergence speed. Bodah *et al.* [13], addressing the particle swarm optimization's propensity to fall into local optima, ingeniously utilized the Levy flight's characteristic of frequent short-distance movements and occasional long-distance jumps to innovatively improve the velocity update formula, effectively enhancing the algorithm's ability to escape local extremum points and strengthening the robustness of global search. Zhang *et al.* [14] enhanced the global search capability of the salp swarm optimization algorithm by introducing global search strategies from the butterfly optimization algorithm, thereby improving the algorithm's optimization performance.

The Arctic Puffin Optimization (APO) algorithm [15] was proposed by Wang *et al.* in September 2024. Originating from the survival strategies of Arctic puffins in nature, the APO algorithm is inspired by their flight and foraging behaviors. The APO algorithm boasts strong optimization capabilities, few parameters, and a simple principle, making it highly competitive in performance compared to other intelligent optimization algorithms. However, it also has certain drawbacks such as susceptibility to local optima, an imbalance between global search and local exploitation, and unstable solution capabilities, thus leaving room for further improvement.

2. Arctic Puffin Optimization

The mathematical model of the APO algorithm is primarily composed of three stages: the population initialization phase, the aerial flight phase, and the underwater foraging phase.

2.1. Initial Population

The specific behaviors of Arctic puffins in the air and on the water form the basis for the design of the APO algorithm. The APO algorithm initializes the population using the following formula:

$$\vec{X}_i^t = rand * (ub - lb) + lb, i = 1, 2, 3, \cdots, N \tag{1}$$

where $\vec{X}_i^t$ represents the position of the ith Arctic Puffin; *rand* is a random number between 0 and 1; *ub* and *lb* represent the upper and lower bounds, respectively; N is the population size.

2.2. Aerial Flight Stage

The Arctic puffins rely on unique flying and foraging strategies to cope with their challenging survival. In their daily lives, they must adapt flexibly between the ocean and the air, meet their nutritional needs, and adjust to different environments. During the aerial foraging phase, the Arctic Puffin employs two key strategies to address different situations, namely the aerial search strategy and the swooping predation strategy.

- The first strategy is aerial search

The aerial search strategy simulates the behavior of the Arctic Puffin searching for suitable foraging waters while in the air, utilizing Levy flight as its powerful wings to change position. When encountering predators such as seagulls, it employs a spiral flight strategy to evade the predators. The following is the position update formula:

$$\vec{Y}_i^{t+1} = \vec{X}_i^t + \left(\vec{X}_i^t - \vec{X}_r^t\right) * L(D) + R \tag{2}$$

$$R = round\left(0.5 * \left(0.5 + rand\right)\right) * \alpha \tag{3}$$

$$\alpha \sim Normal(0,1) \tag{4}$$

where r is a random integer between 1 and $N-1$, excluding i; $\vec{X}_i^t$ represents the current ith candidate solution in the population; $\vec{X}_r^t$ is a candidate solution randomly selected from the current population, with $\vec{X}_i^t \neq \vec{X}_r^t$; $L(D)$ denotes a random number generated through Levy flight; D is the dimensionality; α is a random number following a standard normal distribution.

- The second strategy is swooping predation

The swooping predation strategy simulates the behavior of the Arctic Puffin rapidly changing its flight direction to dive and feed upon spotting prey. To ensure their survival, they must ensure faster and more successful capture of their prey. To simulate this diving behavior, the algorithm introduces a velocity coefficient S to adjust the displacement of the Arctic Puffin during the dive process. The

following is the position update formula:

$$\vec{Z}_i^{t+1} = \vec{Y}_i^{t+1} * S \tag{5}$$

$$S = \tan\left(\left(rand - 0.5\right) * \pi\right) \tag{6}$$

To achieve the best results under various conditions, the algorithm combines the candidate positions generated by the two strategies, sorts them based on their fitness values, and selects the top N individuals with the best fitness values to form a new population. The equation describing this process is as follows:

$$\vec{P}_i^{t+1} = \vec{Y}_i^{t+1} \bigcup \vec{Z}_i^{t+1} \tag{7}$$

$$new = sort\left(\vec{P}_i^{t+1}\right) \tag{8}$$

$$\vec{X}_i^{t+1} = new\left(1:N\right) \tag{9}$$

2.3. Underwater Foraging Stage

The survival strategy of the Arctic puffin involves two crucial aspects: aerial flight and underwater foraging. The underwater foraging phase consists of three main strategies, each employed under specific circumstances to enhance predation efficiency. These three strategies are gathering foraging, intensifying search, and avoiding predators.

- The first strategy is gathering foraging

The gathering foraging strategy simulates the cooperative foraging behavior of Arctic puffins, and the following equation describes the location update:

$$\vec{W}_i^{t+1} \begin{cases} \vec{X}_{r_1}^t + F * L\left(D\right) * \left(\vec{X}_{r_2}^t - \vec{X}_{r_3}^t\right) & rand \geq 0.5 \\ \vec{X}_{r_1}^t + F * \left(\vec{X}_{r_2}^t - \vec{X}_{r_3}^t\right) & rand < 0.5 \end{cases} \tag{10}$$

where F represents the cooperative factor, adjusting the predation behavior of Arctic puffins. In this paper, $F = 0.5$. The variables r_1, r_2, r_3 are random integers between 1 and $N - 1$ (excluding i), and $\vec{X}_{r_1}^t, \vec{X}_{r_2}^t, \vec{X}_{r_3}^t$ are candidate solutions randomly selected from the current population, and $r_1 \neq r_2 \neq r_3$, $\vec{X}_{r_2}^t \neq \vec{X}_{r_3}^t$.

In equation (10), when $rand < 0.5$ is present, it represents the cooperative foraging behavior of the Arctic Puffin with other members, utilizing a cooperation factor F and engaging in random motion to explore the surrounding environment. When $rand \geq 0.5$ is present, it signifies a more complex food search strategy where the Arctic Puffin initially follows other members and, upon discovering a school of fish, quickly swims to join a more advantageous predatory group.

- The second strategy is intensifying search

The intensified search strategy describes a situation where, as predation proceeds, the food resources in the current foraging area gradually become depleted. To continue meeting their needs, Arctic Puffins must change their underwater position to seek out new food sources. The position update equation for this phase is as follows:

$$\vec{Y}_i^{t+1} = \vec{W}_i^{t+1} * \left(1 + f\right) \tag{11}$$

$$f = 0.1*(rand-1)*\frac{(T-t)}{T} \tag{12}$$

where f is an adaptive factor used to adjust the position of the Arctic puffin in the water. T represents the total number of iterations, and t denotes the current iteration count.

- The third strategy is avoiding predators

The avoiding predator strategy is used to describe the behavior of an Arctic puffin when it spots a nearby predator, and here is the location update equation used for this strategy:

$$\vec{Z}_i^{t+1}\begin{cases}\vec{X}_{r_1}^t + F*L(D)*\left(\vec{X}_{r_1}^t - \vec{X}_{r_2}^t\right) & rand \geq 0.5\\ \vec{X}_{r_1}^t + \beta*\left(\vec{X}_{r_1}^t - \vec{X}_{r_2}^t\right) & rand < 0.5\end{cases} \tag{13}$$

where β is a uniformly distributed number between 0 and 1.

To achieve the best results under various conditions, the algorithm merges the candidate positions generated by the three strategies, sorts them based on their fitness values, and selects the top N individuals with the superior fitness values to form a new population. The equation describing this process is as follows:

$$\vec{P}_i^{t+1} = \vec{W}_i^{t+1} \cup \vec{Y}_i^{t+1} \cup \vec{Z}_i^{t+1} \tag{14}$$

$$new = sort\left(\vec{P}_i^{t+1}\right) \tag{15}$$

$$\vec{X}_i^{t+1} = new(1:N) \tag{16}$$

2.4. Behavior Conversion Factor *B*

The APO algorithm favors aerial foraging for global search in the early stages of iteration and underwater foraging for local exploitation in the later stages. This search mechanism is derived from the natural behavior of the Arctic Puffin, which initially searches for suitable foraging waters in the air and then primarily forages underwater. To achieve a smooth transition from global search to local exploitation, the algorithm incorporates a behavior conversion factor, denoted as B. Here is its definition:

$$B = 2*\log\left(\frac{1}{rand}\right)*\left(1-\frac{t}{T}\right) \tag{17}$$

The value of B can be dynamically adjusted as the iterations progress to accommodate the exploration needs at different stages. Additionally, a parameter C is introduced within the algorithm to determine the strategy to be executed at the current iteration stage by comparing the values of B and C. In the paper, the parameter C is set to 0.5.

3. Improved Arctic Puffin Optimization Algorithm

3.1. The Elite Reverse Learning Strategy

The Elite Reverse Learning Strategy [16] significantly enhances the performance of optimization algorithms by introducing elite particles and a reverse learning

mechanism. This strategy endows algorithms with stronger global search capabilities and higher solution accuracy when tackling complex optimization problems and has been widely applied to the improvement of various optimization algorithms.

The optimization performance of an algorithm is greatly influenced by the quality of the initial solutions; a high-quality initial population can accelerate the convergence of the algorithm and facilitate the discovery of the global optimum. However, the APO algorithm uses random initialization for population, which can lead to poor population diversity and slow convergence. To address this issue, this paper applies the Elite Reverse Learning Strategy to the population initialization phase of the algorithm to improve the quality of initial solutions and enhance global search capabilities.

The basic steps for initializing the population using the Elite Reverse Learning Strategy are as follows:

1) Randomly initialize the population S, and select the top $N/2$ individuals with better fitness values to form the elite population E;

2) Determine the reverse population OE of the elite population E;

3) Merge the populations S and OE to form a new population, and select N individuals with better fitness values to constitute the initial population.

3.2. Tangential Flight Strategy

The Tangent Search Algorithm (TSA) [17], proposed in 2021, is a novel optimization algorithm that introduces a new step size based on a tangent function, denoted as $step * \tan(\theta)$, which is akin to the Levy flight function and is referred to as tangent flight. The search equation that combines global and local wandering for the tangent flight function is as follows:

$$X^{t+1} = X^{t} + step \times \tan(\theta) \tag{18}$$

In this paper, the tangent flight function is used to replace the levy flight function in the original APO algorithm, and Equation (2) becomes (19):

$$\vec{Y}_i^{t+1} = \vec{X}_i^{t} + \left(\vec{X}_i^{t} - \vec{X}_r^{t}\right) * T(D) + R \tag{19}$$

Equation (10) becomes (20):

$$\vec{W}_i^{t+1} \begin{cases} \vec{X}_{r_1}^{t} + F * T(D) * \left(\vec{X}_{r_2}^{t} - \vec{X}_{r_3}^{t}\right) & rand \geq 0.5 \\ \vec{X}_{r_1}^{t} + F * \left(\vec{X}_{r_2}^{t} - \vec{X}_{r_3}^{t}\right) & rand < 0.5 \end{cases} \tag{20}$$

Equation (13) becomes (21):

$$\vec{Z}_i^{t+1} \begin{cases} \vec{X}_{r_1}^{t} + F * T(D) * \left(\vec{X}_{r_1}^{t} - \vec{X}_{r_2}^{t}\right) & rand \geq 0.5 \\ \vec{X}_{r_1}^{t} + \beta * \left(\vec{X}_{r_1}^{t} - \vec{X}_{r_2}^{t}\right) & rand < 0.5 \end{cases} \tag{21}$$

Optimizing the step size is also crucial for algorithm optimization; a large step size is conducive to exploration, while a small step size is beneficial for exploitation. **Figure 1** and **Figure 2** illustrate the random walks of 1000 Levy flights and tangent flights simulated using the Mantegna method [18].

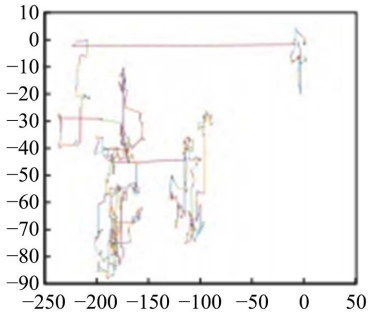


Figure 1. Schematic diagram of the Levy flight random walk.

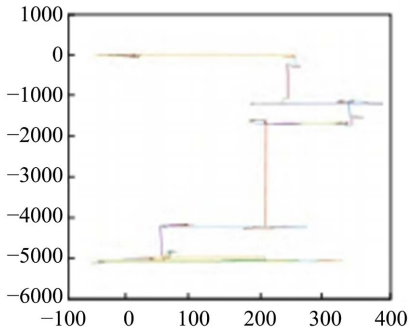


Figure 2. Schematic diagram of the random walk of tangent flight.

From the figures, it can be observed that the step sizes generated by Levy flight exhibit poor randomness and a narrow range, which can lead to the algorithm searching with too small a distance in the early iterations and too large a distance in the later iterations. This results in prolonged optimization iteration cycles and insufficient precision. In contrast, tangent flight has a higher probability of producing large steps and a lower probability of producing small steps. This indicates that tangent flight avoids the issues of search distances being either too large or too small. Therefore, tangent flight is more conducive to helping the algorithm escape from local optima and conduct extensive searches, thereby providing the Arctic Puffin with more opportunities for predation.

The reasons for choosing the tangent flight function over other strategies are:

- The tangent flight function is designed in the Tangent Search Algorithm (TSA) to balance exploration and exploitation, allowing the algorithm to find a good balance between exploration and utilization. This balance helps avoid getting trapped in local optima and enables effective exploration of the entire search space.
- The parameters are simple and flexible, making the algorithm highly adaptable and versatile for different problems.
- The adaptive mechanism allows the algorithm to dynamically adjust the step size at different stages, thereby improving search efficiency.

Thus, replacing the Levy flight function in the original APO algorithm with the tangent flight function allows the improved algorithm to not only avoid insufficient local exploitation but also to further enhance the global search capability of the APO algorithm.

3.3. Adaptive *t*-Distribution Variation Strategy

The *t*-distribution, also known as the student distribution, has a probability density function of:

$$p(x)=\frac{\Gamma\left(\frac{m+1}{2}\right)}{\sqrt{m\pi}\Gamma\left(\frac{m}{2}\right)}\left(1+\frac{x^2}{2}\right)^{-\frac{m+1}{2}} \tag{22}$$

In the formula, Γ represents the gamma function, m is the degrees of freedom parameter, and x is the random variable. The degrees of freedom parameter m influence the shape of the curve. As m approaches infinity, the curve manifests as a Gaussian distribution $N(0,1)$. When m approaches 1, the curve resembles a Cauchy distribution $C(0,1)$. The Gaussian and Cauchy distributions are two boundary special cases of the *t*-distribution. The density function distributions of the three are illustrated in Figure 3.

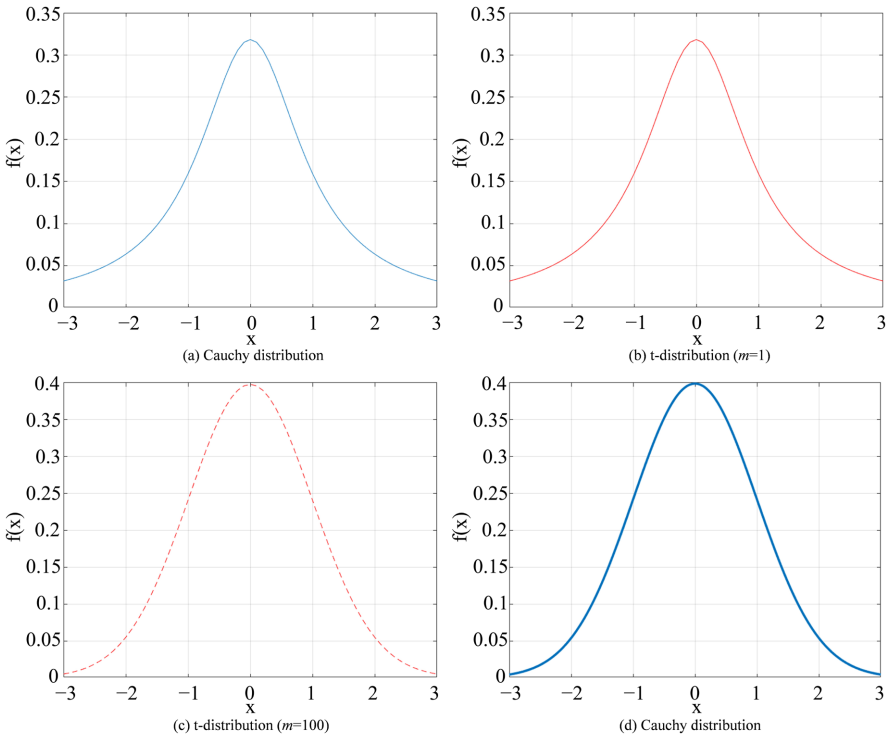


Figure 3. Density function distribution of Gaussian, Cauchy, and *t*-distribution.

The introduction of Gaussian and Cauchy mutations in algorithms has been proven to effectively enhance their performance. Since the Gaussian and Cauchy distributions are two special forms of the *t*-distribution, incorporating *t*-distribution mutation into algorithms can combine the advantages of both Gaussian and Cauchy mutations. Therefore, an adaptive *t*-distribution mutation strategy as shown in Equation (24) is applied to the positions of individuals during the Arctic Puffin's predator avoidance phase:

$$Z_{new} = X_i + X_i \times t(A) \tag{23}$$

$$\vec{Z}_i^{t+1} = \begin{cases} \vec{Z}_i^{t+1}, rand < 0.5 \\ Z_{new}, rand \geq 0.5 \end{cases} \tag{24}$$

here, Z_{new} denotes the new position after t-distribution mutation, and $t(A)$ is the t-distribution with the number of iterations of the APO algorithm as its degrees of freedom. In the early stages of iteration, when the number of iterations is small, the t-distribution mutation resembles Cauchy mutation, which helps maintain population diversity and enhances the algorithm's global exploration capability. In the later stages of iteration, when the number of iterations is large, the t-distribution mutation resembles Gaussian mutation, which is conducive to fine and stable local exploration around the optimal solution, thereby improving convergence accuracy. The flowchart of ETAAPO algorithm as shown in Figure 4.

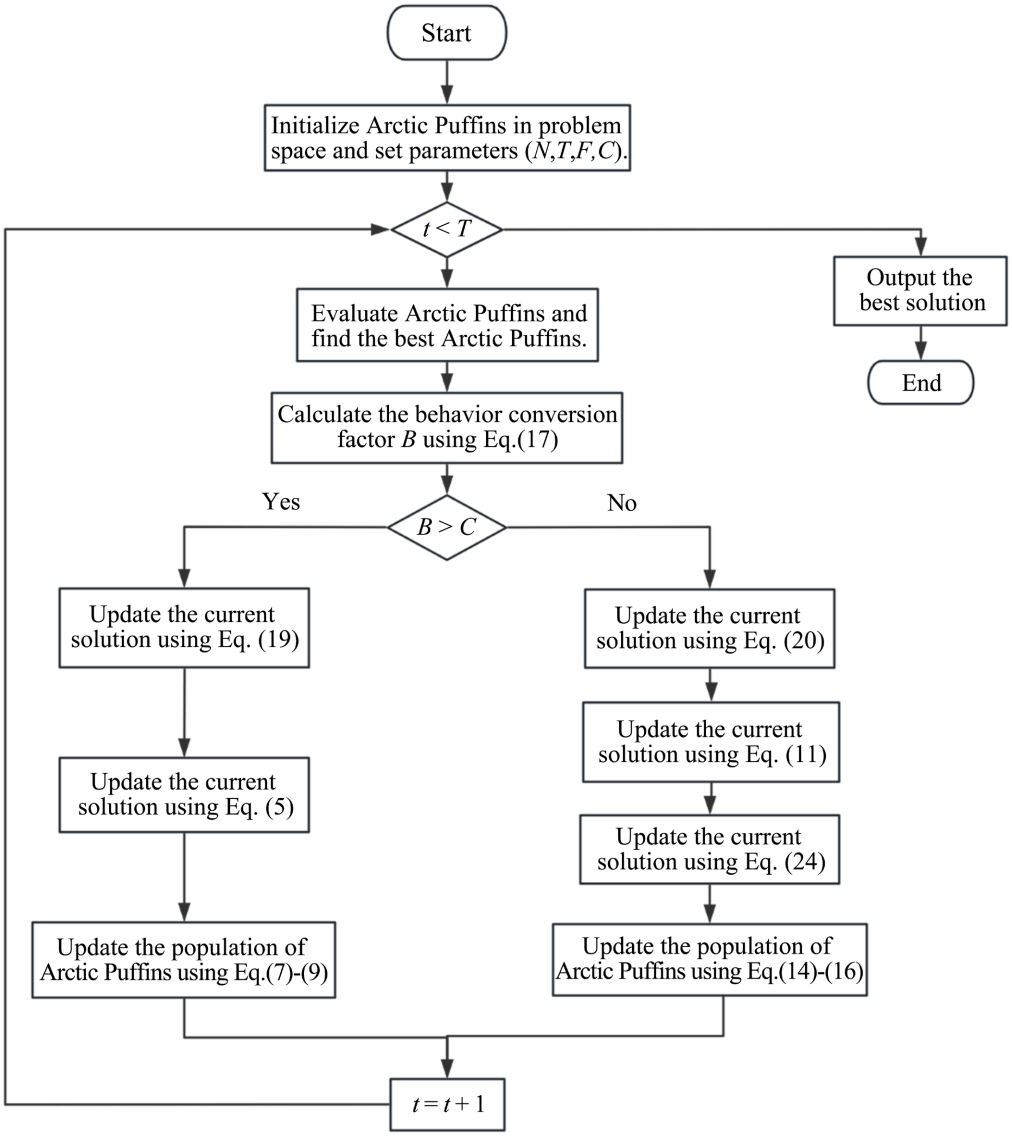


Figure 4. Flowchart of ETAAPO algorithm.

3.4. The Pseudo Code of the ETAAPO

The pseudo code of the ETAAPO is described as follows:

Pseudo (ETAAPO)

Input: N, T, D, F and C
Output: the best $\vec{X}_i^{t+1}$ and its fitness value
initialization the population
define initial parameter (N, T, F, C)
while ($t < T$)
  Calculate the behavior factor B using Eq. (2)
  if $B > C$
    for (i = 1:N)
      update the current solution using Eq. (19)
      update the current solution using Eq. (5)
      Select N excellent populations as the new population $\vec{X}_i^{t+1}$ using Eqs. (7)-(9)
    end for
else
    for (i = 1:N)
      update the current solution using Eq. (20)
      update the current solution using Eq. (11)
      update the current solution using Eq. (24)
      Select N excellent populations as the new population $\vec{X}_i^{t+1}$ using Eqs. (14)-(16)
      Evaluate the puffins, $\vec{X}_i^{t+1}$, and replace $\vec{X}_i^{t}$ with, $\vec{X}_i^{t+1}$ is the better
      $t = t + 1$
    end for
  end if
end while
return $\vec{X}_i^{t+1}$

3.5. Time Complexity Analysis

Time complexity is an important measure of an algorithm's operational efficiency. The time complexity of an algorithm primarily depends on the population size (N), the maximum number of iterations (T), and the dimension (dim). In the APO algorithm, there is an outer loop that runs T times, and within each iteration, an inner loop performs $O(N+\dim)$ operations for each individual, plus an $O(N)$ operation for updating the global optimum. Therefore, the overall time complexity of the APO algorithm is $O(T*N*(N+\dim))$.

In the ETAAPO algorithm, the time complexity of the population initialization phase is $O(N+\dim)$. After initializing the population, the algorithm needs to find the global optimum within the current population, which requires traversing the entire population to determine the best fitness value, making the time complexity of this step $O(N)$. The core of the algorithm consists of T iterations, with each iteration including a traversal of N individuals, thus the time complexity for this part is $O(T*N)$. At the end of each iteration, the algorithm needs to update the global optimum. The time complexity of this step is $O(N)$ as it requires

traversing the entire population to identify the best fitness value. Combining the above analysis, the overall time complexity of the ETAAPO algorithm is $O\left(T*N*\left(N+\dim\right)\right)$.

Therefore, the ETAAPO algorithm proposed in this paper does not increase the time complexity and remains consistent with the APO algorithm.

4. Simulation Experiments and Results Analysis

4.1. Simulation Experiment Environment

The simulation experimental environment for this study is as follows: the operating system is Windows 10, 64-bit operating system, the processor is AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz, the memory is 16.0GB, and the simulation software is MATLAB R2022a.

4.2. Test the Function

To thoroughly validate the performance of the ETAAPO algorithm, this study selects the Grey Wolf Optimizer (GWO) [19], Whale Optimization Algorithm (WOA) [20], Harris Hawks Optimization (HHO) [21], and the newly published Rime optimization algorithm (RIME) [22] in 2024, which have broad application scopes and good performance in recent years, as comparative algorithms based on the CEC2021 benchmark functions. The ETAAPO algorithm is compared with the original Arctic Puffin Optimization (APO) algorithm and the aforementioned comparative algorithms.

The unimodal test benchmark functions within the test suite are used to assess the local exploitation capabilities of the algorithms, while the multimodal test functions are employed to evaluate their global search capabilities. The specific function details are presented in **Table 1**.

Table 1. CEC2021 benchmark functions.

Function	CEC2021	
	name	$f_{\min}$
F1	shifted and rotated bent cigar function	100
F2	shifted and rotated Schwefel's function	1100
F3	shifted and rotated lunacek bi-rastrigin function	700
F4	expanded Rosenbrock's plus Griewangk's function	1900
F5	hybird function 1	1700
F6	hybird function 2	1600
F7	hybird function 3	2100
F8	composition function 1 (N = 3)	2200
F9	composition function 2 (N = 4)	2400
F10	composition function 2 (N = 5)	2500
Range	[−100, 100]	

4.3. Comparative Analysis of the Experimental Results

Table 2. CEC2021 comparison of test function optimization results.

Function	Stats	GWO	WOA	HHO	RIME	APO	ETAAPO
F1	min	3.02E−40	3.14E−93	2.93E−117	6.16E+04	0.6627	**2.05E−256**
	std	1.34E−37	8.14E−75	5.91E−92	1.57E+05	4.1965	**0**
	avg	7.96E−38	1.58E−75	1.08E−92	2.17E+05	5.1174	**2.05E−237**
F2	min	2.87E−38	7.84E−85	2.39E−102	1.68E+05	3.9132	**2.91E−246**
	std	6.21E−37	4.46E−74	3.24E−91	8.77E+05	18.4295	**6.14E−236**
	avg	1.82E−12	0	0	15.3475	260.2252	**0**
F3	min	142.8004	573.8061	0	204.4301	474.7374	**0**
	std	36.6794	150.1930	0	336.0613	2.03E+03	**0**
	avg	1.3118	0	0	352.3584	2.04E+03	**0**
F4	min	776.2944	2.45E+03	0	886.5738	2.60E+03	**0**
	std	0	0	0	15.2339	45.9227	**0**
	avg	57.8367	12.6166	0	10.2774	17.5953	**0**
F5	min	63.5773	2.3035	0	48.0059	90.0743	**0**
	std	75.6839	0	0	47.3602	90.0400	**0**
	avg	185.3820	69.1040	0	68.6764	134.0148	**0**
F6	min	0	0	0	2.4509	5.5541	**0**
	std	1.2481	0.1669	0	0.9749	0.9668	**0**
	avg	0.6094	0.0345	0	3.8099	8.3090	**0**
F7	min	0.0781	0	0	3.5886	8.5715	**0**
	std	5.1828	0.9094	0	6.8215	9.7805	**0**
	avg	3.83E−23	2.73E−84	3.02E−107	190.1351	22.4542	**2.95E−219**
F8	min	5.4379	1.23E−22	6.46E−88	279.9718	103.4684	**5.60E−128**
	std	2.8957	4.90E−23	1.69E−88	771.9838	135.2563	**1.26E−128**
	avg	0.1200	3.85E−71	1.04E−96	763.4462	109.0658	**1.50E−182**
F9	min	26.1986	5.26E−22	3.33E−87	1.34E+03	333.4823	**2.99E−127**
	std	0.0723	0.0098	**−2.22E−16**	6.0269	1.8839	**−2.22E−16**
	avg	6.3075	58.2043	4.90E−04	64.9233	25.7704	**3.38E−17**
F10	min	3.2727	11.1101	1.25E−04	59.0575	16.2282	**−2.10E−16**
	std	0.9594	0.2935	3.34E−08	22.5467	9.4593	**−2.22E−16**
	avg	30.8849	319.2658	0.0026	254.0639	141.5075	**−1.09E−16**

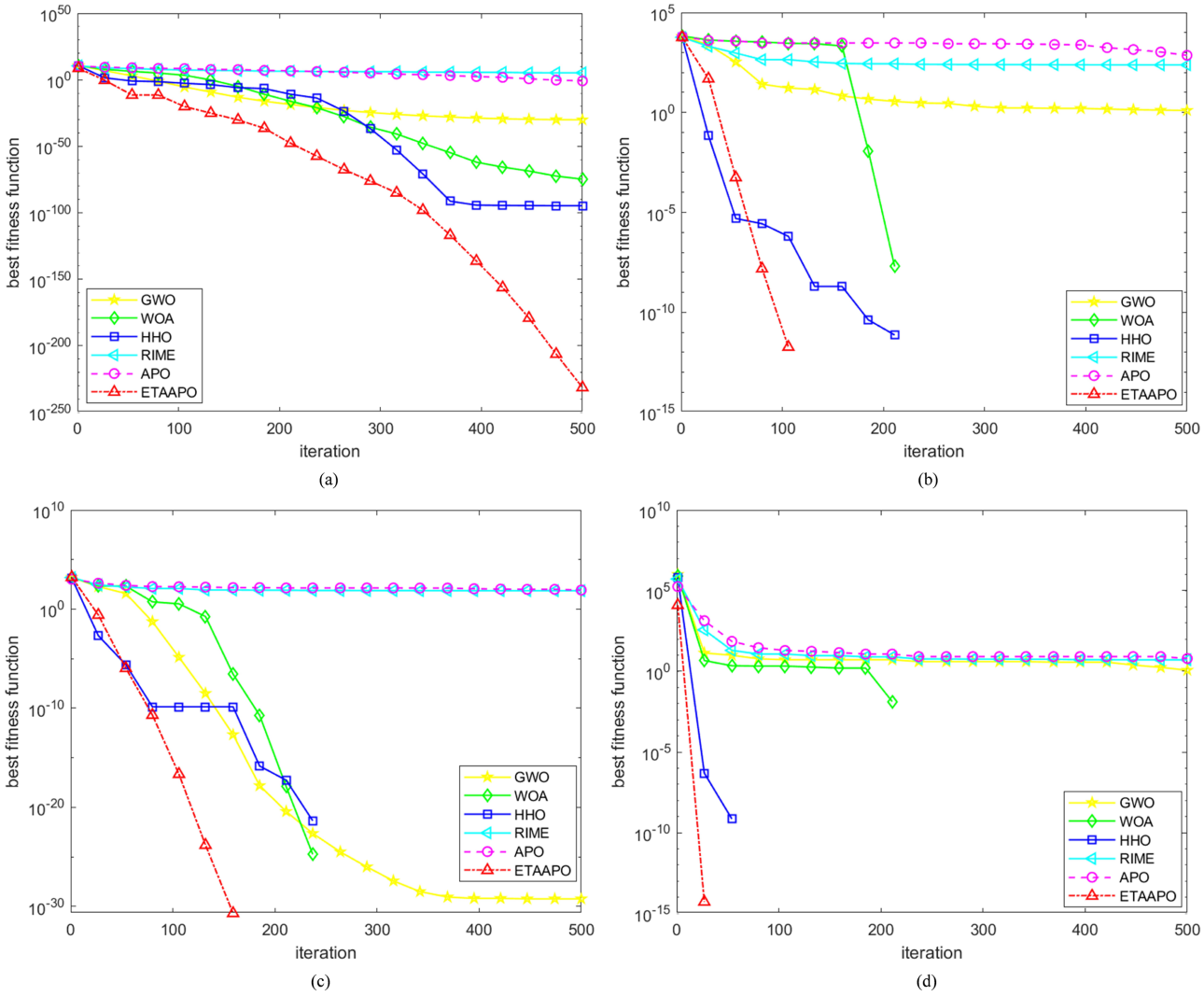
The ETAAPO algorithm proposed in this paper was performance tested against APO, GWO, WOA, HHO, and RIME on the ten benchmark test functions listed in Table 1. In MATLAB R2022a, each experiment was independently executed 30 times with a dimensionality setting of 20 and an iteration count of 500. The

evaluation criteria were based on the optimal values, standard deviations, and average values of the results.

Table 2 compiles the test results for the six algorithms, with the best results highlighted in bold. Correspondingly, the average fitness convergence curves for each algorithm are depicted in Figure 5.

The test results presented in Table 2 demonstrate that the ETAAPO algorithm achieved a 100% optimization effectiveness on functions F3 to F7. Although it did not directly find the optimal values when solving other functions, its optimization accuracy still surpassed that of other comparative algorithms. Compared to other comparative algorithms, the ETAAPO algorithm exhibited a standard deviation of 0 on functions F1, F3 to F7, indicating its strong robustness. Additionally, whether for unimodal or multimodal test functions, the ETAAPO outperformed the other five algorithms not only in terms of optimization accuracy but also in stability.

4.4. Convergence Curve Analysis



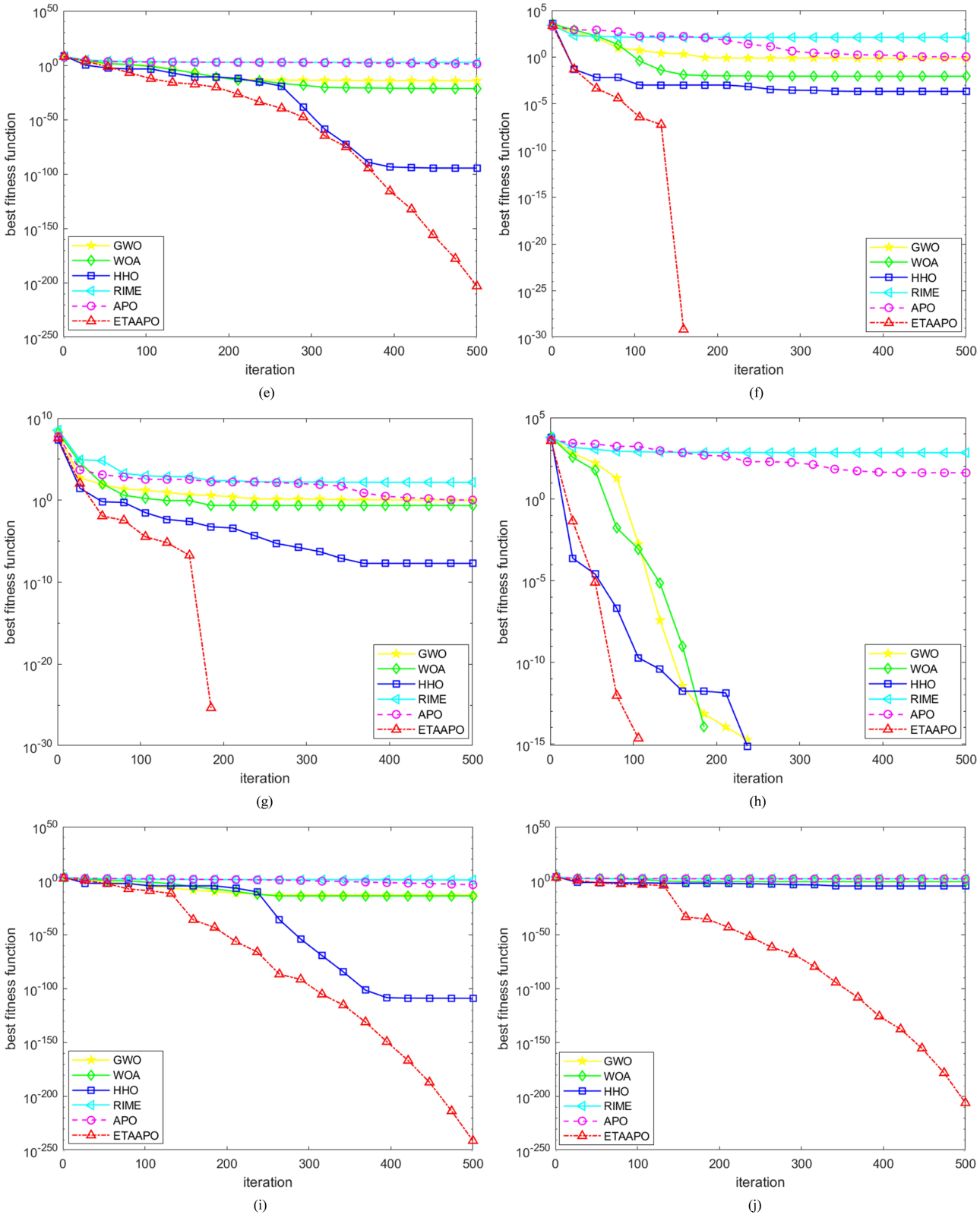


Figure 5. Algorithm convergence curve.

The performance of an algorithm can be intuitively demonstrated through its

convergence curves, which showcase the convergence speed and stability of the algorithm. Figures 5(a)-(j) present a comparison of the convergence curves for six algorithms, including GWO, WOA, HHO, RIME, APO, and ETAAPO, when applied to the aforementioned ten benchmark functions under a 20-dimensional setting. Observing the convergence curves of the aforementioned algorithms, it can be seen that the convergence curve of ETAAPO declines faster than the other five algorithms, and its convergence accuracy is also the best among these six algorithms. This not only indicates that ETAAPO converges faster and has better global search capabilities than the other algorithms, but also that it is less likely to get trapped in local optima, balancing global search capabilities and local development capabilities.

4.5. Wilcoxon Rank-Sum Test

When assessing the performance of an algorithm, it is insufficient to rely solely on the best values, standard deviations, and average values to measure the performance of the improved algorithm. Further statistical testing is required to demonstrate the effectiveness of the Arctic Puffin Optimization algorithm improved with a mixture of strategies and to prove its significant advantages over other existing algorithms. Therefore, this paper employs the Wilcoxon rank-sum test [23] at a significance level of p = 5% and with a dimension of 20.

Table 3. Wilcoxon rank-sum test results.

Function	WOA		GWO		HHO		RIME		APO	
	p	*h*	*p*	*h*	*p*	*h*	*p*	*h*	*p*	*h*
f_1	3.0199e−11	1	3.0199e−11	1	3.0199e−11	1	3.0199e−11	1	3.0199e−11	1
f_2	0.010994	1	1.2019e−12	1	NaN	0	0.010994	1	1.2019e−12	1
f_3	0.16074	0	1.2118e−12	1	NaN	0	0.16074	0	1.2118e−12	1
f_4	0.021577	1	5.772e−11	1	NaN	0	0.021577	1	5.772e−11	1
f_5	3.0199e−11	1	3.0199e−11	1	3.0199e−11	1	3.0199e−11	1	3.0199e−11	1
f_6	3.018e−11	1	3.018e−11	1	2.9803e−11	1	3.018e−11	1	3.018e−11	1
f_7	3.0199e−11	1	3.0199e−11	1	3.0199e−11	1	3.0199e−11	1	3.0199e−11	1
f_8	0.33371	0	NaN	0	NaN	0	0.33371	0	NaN	0
f_9	1.9545e−11	1	1.5262e−11	1	3.0199e−11	1	1.9545e−11	1	1.5262e−11	1
f_{10}	3.0199e−11	1	3.0199e−11	1	3.0199e−11	1	3.0199e−11	1	3.0199e−11	1
1/=/0	8/0/2		9/1/0		6/4/0		10/0/0		10/0/0	

Table 3 presents the results of the Wilcoxon rank-sum test comparing the optimal values obtained from 30 independent runs of the ETAAPO algorithm with those of the Whale Optimization Algorithm (WOA), Grey Wolf Optimizer (GWO), Harris Hawk Optimizer (HHO), Recursive Interdiction Model for Energy (RIME), and the original APO algorithm. The p-value indicates the test

result, and h denotes the significance judgment outcome. When $p < 0.05$, it signifies that the ETAAPO algorithm outperforms the compared algorithm; when $p > 0.05$, it indicates that the ETAAPO algorithm performs worse than the compared algorithm; "NaN" implies inapplicability, meaning that a significance judgment cannot be made. From the statistical results in the table, it can be observed that for most of the 10 benchmark test functions, the p-values are less than 0.05, indicating a significant difference between the ETAAPO and the compared algorithms, with the ETAAPO demonstrating markedly superior performance.

4.6. Engineering Problem and Results Discussion

To validate the avant-garde nature of the ETAAPO algorithm and its superiority in practical engineering applications, this study selects the gear reducer design problem [24] for comparison with several improved algorithms that have demonstrated significant enhancements in recent years. These include the Adaptive Spiral Flight Sparrow Search Algorithm (ASFSSA) [25], the Grey Wolf Optimizer with enhanced convergence factors and proportional weights (CGWO) [26], the Nonlinear Chaotic Harris Hawk Optimization (NCHHO) [27], and the original APO algorithm.

This problem is a relatively classic engineering optimization design challenge, where the optimization objective is to identify a set of seven decision variables that satisfy a range of constraints, including gear bending stress, contact stress, shaft torsional deformation, and stress. The decision variables are as follows: gear width (x_1), gear modulus (x_2), number of teeth on the small gear (x_3), length of bearing 1 (x_4), length of bearing 2 (x_5), diameter of shaft 1 (x_6), and diameter of shaft 2 (x_7), with the goal of minimizing the weight of the reducer. This problem is a mixed-integer programming problem, with variable (x_3) being an integer and all other variables being continuous. The mathematical model of this problem is described as follows:

$$\begin{aligned}\min f(x) &= 0.7854x_1x_2^2\left(3.3333x_3^2+14.9334x_3-43.0934\right)-1.508\left(x_6^2+x_7^3\right)\\ &\quad +7.4777\left(x_6^3+x_7^3\right)+0.7854\left(x_4x_6^2+x_5x_7^2\right)\end{aligned}$$

$$\text{s.t.}\quad g_1(x)=\frac{27}{x_1x_2^2x_3}-1\le 0$$

$$g_2(x)=\frac{397.5}{x_1x_2^2x_3^2}-1\le 0$$

$$g_3(x)=\frac{1.93x_4^3}{x_2x_6^4x_3}-1\le 0$$

$$g_4(x)=\frac{1.93x_5^3}{x_2x_7^4x_3}-1\le 0$$

$$g_5(x)=\frac{\left[\left(745x_4/x_2x_3\right)^2+16.9\times10^6\right]^{1/2}}{110.0x_6^3}-1\le 0$$

$$g_6(x)=\frac{\left[\left(745x_5/x_2x_3\right)^2+157.5\times10^6\right]^{1/2}}{85.0x_7^3}-1\le 0$$

$$g_7(x)=\frac{x_2x_3}{40}-1\le 0$$

$$g_8(x)=\frac{5x_2}{x_1}-1\le 0$$

$$g_9(x)=\frac{x_1}{12x_2}-1\le 0$$

$$g_{10}(x)=\frac{1.5x_6+1.9}{x_4}-1\le 0$$

$$g_{11}(x)=\frac{1.1x_7+1.9}{x_5}-1\le 0$$

$$2.6\le x_1\le 3.6$$

$$0.7\le x_2\le 0.8$$

$$17\le x_3\le 28$$

$$7.3\le x_4, x_5\le 8.3$$

$$2.9\le x_6\le 3.9\ 5.0\le x_7\le 5.5$$

Table 4. Experimental results of reducer problems.

Algorithm	Results		
	Min	Avg	Std
ASFSSA	2999.2989	3.0068e+03	6.9048
CGWO	3286.0921	3.8369e+96	8.9462e+96
NCHHO	3000.0966	3.0177e+03	16.1871
APO	3000.2771	3.0029e+03	1.7754
ETAAPO	**2997.1834**	**3.0008e+03**	**1.4746**

The solution results are presented in Table 4, which compares the outcomes of the ETAAPO algorithm with those of three other improved algorithms and the original algorithm. From the table, it can be observed that in solving the gear reducer design problem, the best values, mean values, and standard deviations of the ETAAPO are all lower than those of the other four improved algorithms. This indicates that the ETAAPO algorithm has higher solution accuracy and better stability in addressing such problems.

5. Conclusions and Implications

Addressing the shortcomings of the Arctic Puffin Optimization (APO) algorithm, such as susceptibility to local optima, imbalance between global search and local exploitation, and unstable solution capabilities, this paper proposes a multi-strategy improved Arctic Puffin Optimization algorithm. By employing an elite reverse

learning strategy for population initialization, the diversity of the population is enhanced, and the convergence speed of the algorithm is accelerated. The original algorithm's Levy flight function is replaced with a tangent flight function, which increases the probability of taking larger steps during random walks, thereby improving the algorithm's global search capability while avoiding insufficient local development. An adaptive t-distribution mutation strategy is utilized to ensure population diversity in the early stages of iteration, which is conducive to global search, and to perform more refined and stable local exploitation in the later stages.

In summary, the Improved Arctic Puffin optimization algorithm (ETAAPO) proposed in this study addresses the limitations of the original APO algorithm by enhancing population diversity, accelerating convergence speed, and improving the balance between global search and local development.

Application results on 10 benchmark test functions, Wilcoxon rank-sum tests, and engineering optimization problems demonstrate that the Enhanced Tangent Arctic Puffin Optimization (ETAAPO) algorithm has faster convergence speed, higher convergence accuracy, and a stronger ability to escape from local optima. Future work will continue to improve the performance of the Arctic Puffin algorithm, enhancing its optimization accuracy, convergence speed, and convergence stability. On this basis, the ETAAPO algorithm will be considered for application in solving complex multi-objective problems and practical engineering case studies.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Zhao, H.W., Liu, Y.Q., Dong, L.Y., *et al.* (2018) Hybrid Dynamic Path Optimization Algorithm for Intelligent Transportation. *Journal of Jilin University* (*Engineering Science*), **48**, 1214-1223.

[2] Liu, R. (2020) Research on Intelligent Optimization Algorithm and Its Application in the Financial Field. Ph.D. Thesis, Jilin University.

[3] Luo, L.J., Zhao, L.P. and Mo, C.K. (2023) Research and Analysis on Optimization of Intelligent Manufacturing Production Line Based on Firefly Algorithm. *Internal Combustion Engine and Accessories*, No. 23, 119-121.

[4] Liu, J.F., Chen, J.L., Wang, X.S., *et al.* (2020) Research on Energy Management and Optimization Strategy of Micro Energy Grid Based on Deep Reinforcement Learning. *Power System Technology*, **44**, 3794-3803.

[5] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002) A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, **6**, 182-197. https://doi.org/10.1109/4235.996017

[6] Kennedy, J. and Eberhart, R. (1995) Particle Swarm Optimization. *Proceedings of ICNN'95—International Conference on Neural Networks*, Perth, 27 November-1 December 1995, 1942-1948. https://doi.org/10.1109/icnn.1995.488968

[7] Arora, S. and Singh, S. (2013) The Firefly Optimization Algorithm: Convergence Analysis and Parameter Selection. *International Journal of Computer Applications*, **69**, 48-52. https://doi.org/10.5120/11826-7528

[8] Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H. and Mirjalili, S.M. (2017) SALP Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems. *Advances in Engineering Software*, **114**, 163-191. https://doi.org/10.1016/j.advengsoft.2017.07.002

[9] Huo, J., Liu, L. and Zhang, Y. (2018) An Improved Multi-Cores Parallel Artificial Bee Colony Optimization Algorithm for Parameters Calibration of Hydrological Model. *Future Generation Computer Systems*, **81**, 492-504. https://doi.org/10.1016/j.future.2017.07.020

[10] Arora, S. and Singh, S. (2018) Butterfly Optimization Algorithm: A Novel Approach for Global Optimization. *Soft Computing*, **23**, 715-734. https://doi.org/10.1007/s00500-018-3102-4

[11] Xue, J. and Shen, B. (2020) A Novel Swarm Intelligence Optimization Approach: Sparrow Search Algorithm. *Systems Science & Control Engineering*, **8**, 22-34. https://doi.org/10.1080/21642583.2019.1708830

[12] Ma, C., Zhou, D.Q. and Zhang, Y. (2020) BP Neural Network Water Resources Demand Forecasting Method Based on Improved Whale Algorithm. *Computer Science*, **47**, 496-500.

[13] Bodha, K.D., Mukherjee, V., Yadav, V.K., Saurabh, K. and Anium, S. (2018) A Levy Flight Based Voltage Particle Swarm Optimization for Multiple-Objective Mixed Cost-Effective Emission Dispatch. 2018 8*th International Conference on Cloud Computing*, *Data Science & Engineering* (*Confluence*), Noida, 11-12 January 2018, 1-6. https://doi.org/10.1109/confluence.2018.8442919

[14] Zhang, L.Z. and Wang, Z.S. (2024) Adaptive Sea Squirt Swarm Algorithm Based on Global Search Strategy. *Computer Simulation*, **41**, 360-368.

[15] Wang, W., Tian, W., Xu, D. and Zang, H. (2024) Arctic Puffin Optimization: A Bio-Inspired Metaheuristic Algorithm for Solving Engineering Design Optimization. *Advances in Engineering Software*, **195**, Article ID: 103694. https://doi.org/10.1016/j.advengsoft.2024.103694

[16] Tizhoosh, H.R. (2005) Opposition-Based Learning: A New Scheme from a China Intelligence. *Proceedings of the International Conference on Computational Intelligence for Modelling*, *Control and Automation and International Conference on International Conference Intelligences*, *Web Technologies and Internet Commerce* (*CIMCA-IAWTIC*'06), Vienna, 28-30 November 2005, 695-701.

[17] Layeb, A. (2022) Tangent Search Algorithm for Solving Optimization Problems. *Neural Computing and Applications*, **34**, 8853-8884. https://doi.org/10.1007/s00521-022-06908-z

[18] Mantegna, R.N. (1994) Fast, Accurate Algorithm for Numerical Simulation of Lévy Stable Stochastic Processes. *Physical Review E*, **49**, 4677-4683. https://doi.org/10.1103/physreve.49.4677

[19] Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014) Grey Wolf Optimizer. *Advances in Engineering Software*, **69**, 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[20] Mirjalili, S. and Lewis, A. (2016) The Whale Optimization Algorithm. *Advances in Engineering Software*, **95**, 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008

[21] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H. (2019) Harris Hawks Optimization: Algorithm and Applications. *Future Generation Computer Systems*, **97**, 849-872. https://doi.org/10.1016/j.future.2019.02.028

[22] Su, H., Zhao, D., Heidari, A.A., Liu, L., Zhang, X., Mafarja, M., *et al.* (2023) RIME: A Physics-Based Optimization. *Neurocomputing*, **532**, 183-214. https://doi.org/10.1016/j.neucom.2023.02.010

[23] Wilcoxon, F. (1945) Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, **1**, 80-83. https://doi.org/10.2307/3001968

[24] Agushaka, J.O., Ezugwu, A.E. and Abualigah, L. (2022) Dwarf Mongoose Optimization Algorithm. *Computer Methods in Applied Mechanics and Engineering*, **391**, Article ID: 114570. https://doi.org/10.1016/j.cma.2022.114570

[25] Ouyang, C., Qiu, Y. and Zhu, D. (2021) Adaptive Spiral Flying Sparrow Search Algorithm. *Scientific Programming*, **2021**, Article ID: 6505253. https://doi.org/10.1155/2021/6505253

[26] Wang, Q.P., Wang, M.N. and Wang, X.F. (2019) Gray Wolf Optimization Al-Gorithm with Improved Convergence Factor and Proportional Weight. *Computer Engineering and Applications*, **55**, 60-65.

[27] Dehkordi, A.A., Sadiq, A.S., Mirjalili, S. and Ghafoor, K.Z. (2021) Nonlinear-based Chaotic Harris Hawks Optimizer: Algorithm and Internet of Vehicles Application. *Applied Soft Computing*, **109**, Article ID: 107574. https://doi.org/10.1016/j.asoc.2021.107574