Postprint of Aquila Optimizer Integrating Differential Mutation and Tangent Flight

Authors: Xu Yifeng, Liu Sheng, Yusong Liu, Zhang Weikang

Date: 2022-06-06T00:00:00+00:00

Abstract

To address the issue that the Aquila Optimizer (AO) possesses strong global exploration capability but insufficient local exploitation capability, this paper proposes the Differential Evolution mutation and tangent flight Aquila Optimizer (DEtanAO). First, the mutation operation from the Differential Evolution algorithm is utilized, which provides the algorithm with strong exploitation capability and compensates for the deficiency of the AO algorithm. Subsequently, the tangent flight strategy from the tangent search algorithm, which possesses strong ability to explore the search space and enables the algorithm to escape local optima, is employed to replace the Lévy flight in the AO algorithm. The combination of these two strategies effectively balances the exploration and exploitation phases of the DEtanAO algorithm. Finally, to verify the optimization performance of the DEtanAO algorithm, the optimization capability of the improved algorithm is tested on 12 standard benchmark functions, high-dimensional functions, Wilcoxon rank-sum test, and engineering optimization problems. Experimental results demonstrate that, compared with other newly proposed intelligent algorithms, the DEtanAO algorithm exhibits stronger optimization capability and faster convergence speed.

Full Text

Preamble

Aquila Optimizer Integrating Differential Mutation and Tangent Flight

Xu Yifeng, Liu Sheng†, Liu Yusong, Zhang Weikang
(School of Management, Shanghai University of Engineering Science, Shanghai 201620, China)

Abstract: The Aquila optimizer (AO), although capable of robust global exploration, suffers from inadequate local exploitation capability. To address this limitation, we propose a differential evolution mutation and tangent flight Aquila optimizer (DEtanAO). First, we incorporate the mutation operation from differential evolution algorithms, which endows the algorithm with strong exploitation ability to compensate for AO' s deficiency. Second, we replace AO' s Lévy flight with the tangent flight strategy from the tangent search algorithm, which possesses superior ability to explore the search space and escape local optima. The combination of these two strategies effectively balances the exploration and exploitation phases of the DEtanAO algorithm. To validate the optimization performance of DEtanAO, we evaluate its search capability on 12 standard benchmark functions, high-dimensional functions, through Wilcoxon rank-sum tests, and on engineering optimization problems. Experimental results demonstrate that DEtanAO exhibits stronger optimization ability and faster convergence speed compared to other recently proposed intelligent algorithms.

Keywords: aquila optimizer; differential evolution mutation; tangent flight; exploration and exploitation

0 Introduction

As real-world optimization problems with discrete and unconstrained properties become increasingly complex, traditional mathematics-based optimization algorithms struggle to find optimal solutions and suffer from notable drawbacks and limitations, such as convergence to local optima, unknown search spaces, and single-solution approaches. To overcome these challenges, bio-inspired metaheuristic optimization algorithms have been continuously proposed and improved in recent years, gaining popularity among researchers due to their operational simplicity and straightforward parameter tuning. Examples include the chaotic particle swarm optimization algorithm (CPSO) that incorporates adaptive weights to simulate bird flock foraging behavior, the sparrow search algorithm (SSA) inspired by sparrow foraging and regurgitation behaviors, the marine predators algorithm (MPA), and the elite opposition-based golden-sine whale optimization algorithm (EGolden-SWOA) that integrates elite opposition and golden sine strategies. These algorithms provide diverse perspectives for solving complex optimization problems.

The Aquila Optimizer (AO), proposed by Laith Abualigah et al. in 2021, is a novel population-based optimization method inspired by the hunting behavior of aquilas in nature. While AO demonstrates powerful global exploration capability, high search efficiency, and fast convergence speed, it suffers from insufficient local exploitation ability and tends to fall into local optima. Wang et al. proposed an improved hybrid Aquila optimizer and Harris Hawks optimizer (IHAOHHO), integrating HHO' s exploitation strategy into AO' s exploration strategy and adding a reverse opposition-based learning (ROBL) mechanism to avoid local optima stagnation. Additionally, a nonlinear escape energy parameter balances the algorithm' s exploration and exploitation phases, enhancing IHAOHHO' s optimization performance. Shubham Mahajan et al. combined AO with the arithmetic optimization algorithm (AO-AOA) to solve optimization problems, validating the algorithm' s effectiveness and superiority across 23 benchmark functions. Shicheng Wang et al. proposed an improved multi-objective Aquila optimizer to design a hybrid system based on solid oxide fuel cells (SOFC), improving overall cost and exergy efficiency. Eric Ofori-Ntow Jnr et al. developed a DWT-PSR-AOA-BPNN model by hybridizing discrete wavelet transform (DWT), phase space reconstruction from chaos theory (PSR), Aquila optimization algorithm (AOA), and back-propagation neural network (BPNN) for wind speed prediction, demonstrating high prediction accuracy. However, since AO was proposed relatively recently, domestic research on its improvement remains insufficient, leaving room for performance enhancement to enable broader practical applications. This paper proposes a DEtanAO algorithm that integrates differential mutation and tangent flight. We first incorporate a differential mutation strategy into AO to improve search precision, then replace AO' s Lévy flight with the tangent flight strategy from the tangent search algorithm (TSA) to further enhance AO' s optimization performance and enable escape from local optima. To demonstrate the effectiveness of the improved algorithm, we conduct comparative experiments with multiple algorithms on unimodal, multimodal, and high-dimensional test functions, perform Wilcoxon rank-sum test analysis, and further validate the algorithm' s advancement through engineering application problems. Results show that DEtanAO achieves significant improvements in optimization accuracy and convergence speed.

1 Aquila Optimizer

The Aquila optimizer simulates different hunting strategies for different prey types. The hunting method for fast-moving prey reflects the algorithm' s global exploration capability, while the approach for slow-moving prey reflects its local exploitation capability. AO exhibits strong global exploration, high search efficiency, and fast convergence, but suffers from insufficient local exploitation and susceptibility to local optima.

AO models aquila behavior during hunting through four strategies:

1. **Expanded Exploration:** High soar with vertical stoop to select search space
2. **Narrowed Exploration:** Contour flight with short glide attack to explore within search space
3. **Expanded Exploitation:** Low flight with slow descent attack to explore within converged search space
4. **Narrowed Exploitation:** Walk and grab prey to dive and attack

1.1 Expanded Exploration

In the first strategy, the aquila identifies prey regions and selects optimal hunting areas through high soaring with vertical stoop. From high altitudes, the aquila determines search space regions–i.e., where prey might be located. [Figure 1: see original paper] illustrates this high soaring and vertical stoop behavior.

This behavior is mathematically represented by Equation (1):

$$X_1(t+1) = X_{best}(t) \times \left(1 - \frac{t}{T}\right) + (X_M(t) - X_{best}(t)) \times \text{rand}$$

where X_1 represents the solution generated by the first search method at iteration $t+1$, $X_{best}(t)$ is the optimal solution at iteration t reflecting the prey' s approximate location, t/T controls exploration through iteration count, $X_M(t)$ denotes the mean of current solutions at iteration t, rand is a random number in [0,1], t and T represent current and maximum iteration numbers respectively, Dim is problem dimension, and N is the number of candidate solutions.

1.2 Narrowed Exploration

In the second strategy, after locating prey regions from high altitude, the aquila performs contour flight with short glide attacks within the search space. This behavior is represented by Equation (3):

$$X_2(t+1) = X_{best}(t) \times \text{Levy}(D) + X_R(t) + y \times (x - \text{rand})$$

where X_2 is the solution generated by the second search method at iteration $t+1$, D represents dimensional space, Levy(D) is the Lévy flight distribution function, and X_R is a random solution with index range [1,N]. The Lévy flight is calculated as:

$$\text{Levy}(D) = s \times \frac{u}{|v|^{1/\beta}}$$

with

$$s = 0.01 \times \frac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma((1+\beta)/2) \times \beta \times 2^{(\beta-1)/2}}$$

The spiral search components are computed as:

$$y = r \times \cos\theta, \quad x = r \times \sin\theta$$

$$r = r_1 + U \times D_1, \quad \theta = -\omega \times D_1 + \theta_1$$

$$\theta_1 = 3 \times \pi/2$$

where β is a constant fixed at 1.5, U and D_1 are integers from 1 to search space length, ω is fixed at 0.00565, and r_1 is 0.005. [Figure 2: see original paper] shows the contour flight with short glide attack behavior.

1.3 Expanded Exploitation

In the third strategy, after locking onto prey regions, the aquila prepares to land and attack through vertical descent and preliminary strikes to test prey response. This low flight with slow descent attack is illustrated in [Figure 3: see original paper] and mathematically expressed by Equation (10):

$$X_3(t+1) = (X_{best}(t) - X_M(t)) \times \alpha - \text{rand} + (\text{UB} - \text{LB}) \times \delta$$

where X_3 is the solution generated by the third search method at iteration $t+1$, α and δ are small exploitation adjustment parameters in (0,1), and LB and UB represent the lower and upper bounds of the problem.

1.4 Narrowed Exploitation

In the fourth strategy, as the aquila approaches prey, it circles above the target and prepares to attack based on the prey' s random movements. This "walk and grab prey" behavior is shown in [Figure 4: see original paper] and represented by Equation (11):

$$X_4(t+1) = QF \times X_{best}(t) - G_1 \times X(t) \times \text{rand} - G_2 \times \text{Levy}(D) + \text{rand} \times G_1$$

where X_4 is the solution generated by the fourth search method at iteration $t+1$, QF is a quality function balancing search strategies:

$$QF(t) = \frac{2 \times \text{rand} \times (1 - t/T)}{1}$$

G_1 represents different methods the aquila employs during prey escape, and G_2 is a decreasing value from 2 to 0 representing the flight slope from first to last position during prey tracking. [Figure 5: see original paper] shows the AO algorithm flowchart.

Both AO phases require greedy algorithms to ensure new positions outperform original ones. Examining AO' s update equations reveals that Equations (1) and (11) converge toward 0, Equation (10) converges to constants, and only Equation (3) can potentially converge to optimal solutions, making AO prone to local optima.

2 Proposed DEtanAO Algorithm

2.1 Differential Mutation Strategy

Differential evolution (DE) is a population-based evolutionary algorithm that iteratively approaches global optima through three operations: mutation, crossover, and selection. DE generates mutant populations through differential mutation, creates crossover populations by crossing mutant and original populations, and selects the next generation via greedy selection. Differential mutation combines mutant individuals with difference vectors derived from two distinct parent vectors. Common mutation strategies are listed in Table 1.

In Table 1, F is the scaling factor, $r_1 \neq r_2 \neq r_3$ are random integers, $X^t_{best,j}$ represents the best individual in generation j, $V^t_{i,j}$ is the mutated individual, $X^t_{i,j}$ is the ith individual in generation j, and $X^t_{r_1,j}$, $X^t_{r_2,j}$, $X^t_{r_3,j}$ are randomly selected distinct individuals from the current population.

The "best" -based mutation strategy offers strong exploitation and rapid convergence. Using the fittest individual from the current aquila population as the base vector, with r_1 and r_2 randomly generated from the population, the mutation factor F varies dynamically with iterations. Early in the search, smaller F values minimize scaling effects, causing mutated aquilas to rapidly approach the population' s best individual. Later, larger F values increase scaling effects, enabling the population to perform fine-grained local exploration around the best individual and improving search precision.

Table 1: Variation Strategies

Strategy	Mathematical Expression
DE/rand/1	$V^t_{i,j} = X^t_{r_1,j} + F \times (X^t_{r_2,j} - X^t_{r_3,j})$
DE/best/1	$V^t_{i,j} = X^t_{best,j} + F \times (X^t_{r_2,j} - X^t_{r_3,j})$
DE/current-to-best/1	$V^t_{i,j} = X^t_{i,j} + F \times (X^t_{best,j} - X^t_{i,j}) + F \times (X^t_{r_1,j} - X^t_{r_2,j})$

2.2 Tangent Flight Strategy

The Tangent Search Algorithm (TSA), proposed in 2021, is a novel population-based optimization algorithm that uses a tangent function-based mathematical model to move solutions toward better ones. Most optimization algorithms, whether derivative-based or metaheuristic, follow a descent equation of the form:

$$X^* = X + \text{step} \times d$$

where step is movement magnitude and d is movement direction. The algorithms differ in step size calculation: derivative-based methods use gradient descent or Hessian matrices, while metaheuristics use random steps. For instance, genetic algorithms use Gaussian mutation, differential evolution uses differences between population individuals, and cuckoo search uses Lévy flight functions.

Step size optimization is critical—large values favor exploration while small values favor exploitation. TSA proposes a novel step size based on the tangent function, called tangent flight, which effectively explores the search space. The combined global-local search equation is:

$$X^* = X + \text{step} \times \tan(\theta)$$

Mathematically, θ values closer to $\pi/2$ produce larger tangent values and solutions farther from the current position, while values near 0 produce smaller tangent values and nearby solutions. The exploration equation applies to each variable with probability $1/D$, where D is problem dimension.

Replacing AO's Lévy flight with tangent flight enhances AO's global exploration while mitigating insufficient local exploitation. [Figure 6: see original paper] and [Figure 7: see original paper] show random walk simulations of Lévy flight and tangent flight using the Mantegna method. Lévy flight exhibits high randomness with narrow step size ranges, potentially causing insufficient search distances early and excessive distances late in iterations, leading to long optimization cycles and low precision. Tangent flight produces large steps more frequently, compensating for Lévy flight' s distance issues and enabling better escape from local optima through broader searches, thus providing aquilas with more hunting opportunities.

2.3 Time Complexity Analysis of DEtanAO

Time complexity analysis typically depends on three components: population initialization, fitness function calculation, and solution updates. With population size N, initialization complexity is $O(N)$. The update process complexity is $O(T \times N) + O(T \times N \times \text{Dim})$, where T is total iterations and Dim is problem dimension. Thus, AO' s total complexity is $O(N \times (T \times D + 1))$.

DEtanAO only adds the differential evolution mutation strategy $O(T \times N)$ and replaces the Lévy flight with tangent flight without increasing computational complexity. Therefore, DEtanAO' s total complexity is $O(N \times (T \times (D + 1) + 1))$, maintaining the same complexity class as standard AO without additional computational overhead. The DEtanAO pseudo-code is presented below.

Algorithm: Pseudo-code of DEtanAO

```
Output: best position, best fitness
Initialize population % Population initialization
Define initial parameters ($\alpha$, $\delta$, etc.) % Define initial parameters
while (termination condition not met) do
    Calculate fitness values and best solution X_{best}(t)
    for i = 1,2,⋯,N do
        Update parameters % Update parameters
        Expanded exploration: Update current solution using Equation (1)
        Narrowed exploration: Update current solution using Equation (3)
        Expanded exploitation: Update current solution using Equation (10)
        Narrowed exploitation: Update current solution using Equation (11)
    end for
    Mutation: Generate mutant population % Differential mutation strategy
```

```
    Tangent flight % Tangent flight strategy
end while
```

3 Experimental Results and Analysis

3.1 Simulation Environment

The simulation environment uses Windows 10 64-bit OS, Intel® CoreTM i5-10210U CPU @ 1.60GHz/2.11 GHz, 16.0GB RAM, and MATLAB 2020b.

3.2 Comparison Algorithms and Parameter Settings

We compare DEtanAO against basic Aquila optimizer (AO), chaotic particle swarm optimization (CPSO), sparrow search algorithm (SSA), marine predators algorithm (MPA), elite opposition-based golden-sine whale optimization algorithm (EGolden-SWOA), and tanAO (AO with tangent flight replacing Lévy flight). For fair comparison, all algorithms use population size 30 and 500 iterations. Other parameters are listed in Table 2.

Table 2: Experimental Parameters of Each Algorithm

Algorithm	Parameters
tanAO	α=0.1, δ=0.1, u=0.0056, ω=0.005
DEtanAO	α=0.1, δ=0.1, u=0.0056, ω=0.005
EGolden-SWOA	C1=1.5, C2=1.5, u=2
SSA	ST=0.8, PD=0.2, SD=0.2
MPA	P=0.5, FADs=0.2

3.3 Test Functions

To validate optimization performance, we select 12 benchmark functions: unimodal functions (f_1-f_6) for local exploitation assessment, multimodal functions (f_7-f_{10}) for global exploration assessment, and fixed-dimension multimodal functions (f_{11}-f_{12}). Details are provided in Table 3.

Table 3: Benchmark Functions

Function	Name	Search Range
f_1	Sphere	[-100,100]
f_2	Schwefel 2.22	[-30,30]
f_3	Schwefel 1.2	[-100,100]
f_4	Schwefel 2.21	[-100,100]
f_5	Rosenbrock	[-30,30]
f_6	Quartic	[-1.28,1.28]
f_7	Rastrigin	[-5.12,5.12]
f_8	Ackley	[-32,32]

Function	Name	Search Range
f_9	Penalized	[-50,50]
f_{10}	Foxholes	[-65,65]
f_{11}	Branin	[-5,5]
f_{12}	Goldstein Price	[-2,2]

3.4 Optimization Accuracy Analysis

We independently run DEtanAO, AO, tanAO, CPSO, SSA, MPA, and EGolden-SWOA 30 times on 12 benchmark functions with dimensions dim=30/100/500 for non-fixed dimension functions. Tables 4 and 5 summarize the best, mean, and standard deviation results for low-dimensional, high-dimensional, and fixed-dimension cases.

Table 4 results show DEtanAO achieves 100% optimization on functions f_1, f_6, f_7, f_9, f_{10}, directly finding global optima. Compared to other algorithms, DEtanAO shows zero standard deviation on functions f_2, f_3, f_4, f_8, f_{11}, indicating strong robustness ensured by differential mutation’ s population diversity. In low dimensions, DEtanAO outperforms competitors by 11 to 235 orders of magnitude. In high dimensions, DEtanAO maintains superior performance on large-scale functions, still finding optimal solutions for f_1 and f_3 while preserving low-dimensional performance levels, demonstrating no “curse of dimensionality” and strong stability.

Table 5 shows that while MPA performs best on fixed-dimension functions, DEtanAO still directly finds optimal solutions with good robustness as indicated by standard deviations. Overall, DEtanAO significantly improves stability and accuracy over other algorithms. The differential mutation strategy reduces AO’ s over-reliance on position update formulas based on attack patterns, while tangent flight enables escape from local optima and improves optimization quality.

3.5 Convergence Curve Analysis

Convergence curves intuitively reveal algorithm performance, showing convergence speed and local optima trapping frequency. Figure 8: see original paper-(i) presents convergence curves for all seven algorithms on 12 benchmark functions at 30 dimensions. DEtanAO consistently converges faster and achieves higher precision than the other six algorithms throughout iterations, demonstrating superior global exploration without falling into local optima, effectively balancing exploration and exploitation. To validate improvement strategies, tanAO (using only tangent flight) shows significantly faster convergence than basic AO—for instance, reaching optimal values in 150 iterations versus AO’ s 250 iterations on some functions, substantially improving both accuracy and speed.

3.6 Wilcoxon Rank-Sum Test

The Wilcoxon rank-sum test is a non-parametric statistical method for determining significant differences between DEtanAO and other algorithms. Table 6 shows results where "p" indicates test results and "h" shows significance judgment. When $p < 0.05$, $h = 1$ indicates DEtanAO is significantly superior; when $p > 0.05$, $h = 0$ indicates inferior significance; "N/A" means no test was possible.

Comparing 30-run mean values, most p-values are below 0.05, confirming significant differences and DEtanAO' s superior performance across multiple functions.

3.7 Engineering Optimization Problem Application

To validate DEtanAO's advancement and practical superiority, we apply it to the pressure vessel design optimization problem and compare with other improved algorithms: SCSSA (sine-cosine and Cauchy mutation SSA), MSSSA (mixed strategy improved SSA), FWOA (whale optimization with piecewise random inertia weight and optimal feedback), and G-QPSO (Gaussian quantum-behaved PSO). The pressure vessel design problem aims to minimize manufacturing cost by optimizing variables: cylindrical length L, inner radius R, cylindrical wall thickness T_s, and head wall thickness T_h.

Table 7 compares results, showing DEtanAO achieves smaller best and mean values than competitors, demonstrating higher solution accuracy and stability for such optimization problems, thus validating the algorithm' s advancement and superiority.

4 Conclusion

This paper proposes DEtanAO, which integrates differential evolution and tangent flight strategies. The differential mutation strategy reduces AO' s over-reliance on position updates based on attack patterns, while tangent flight increases the probability of large steps during random walks, enabling escape from local optima and improving optimization quality. Results from 12 standard benchmarks, high-dimensional functions, Wilcoxon rank-sum tests, and engineering applications demonstrate DEtanAO' s strong ability to escape local optima, faster convergence, and higher precision. Future work will apply DEtanAO to other practical problems, such as hyperparameter optimization in machine learning prediction and classification algorithms, and further apply the optimized machine learning algorithms to predict stock price trends in quantitative investment, expanding the algorithm' s applications in finance and other domains.

References

[1] Yu Z, Li Q, Feng Y, et al. Hierarchical economic load dispatch based on chaotic-particle swarm optimization [C]// Ninth International Conference on Natural Computation. IEEE, 2013: 517-521.

[2] Xue J, Shen B. A novel swarm intelligence optimization approach: sparrow search algorithm [J]. Systems Science & Control Engineering An Open Access Journal, 2020, 8(1): 22-34.

[3] Faramarzi A, Heidarinejad M, Mirjalili S, et al. Marine predators algorithm: A nature-inspired metaheuristic [J]. Expert Systems with Applications, 2020, 152: 113377.

[4] Xiao Z, Liu S. Study on elite opposition-based golden-sine whale optimization algorithm and its application of project optimization [J]. Acta Electronica Sinica, 2019, 47(10): 2177-2186.

[5] Abualigah L, Yousri D, Elaziz M A, et al. Matlab Code of Aquila Optimizer: A novel meta-heuristic optimization algorithm [J]. Computers & Industrial Engineering, 2021, 157: 107250.

[6] Wang S, Jia H, Abualigah L, et al. An Improved Hybrid Aquila Optimizer and Harris Hawks Algorithm for Solving Industrial Engineering Optimization Problems [J]. Processes, 2021, 9(9): 1551.

[7] Mahajan S, Abualigah L, Pandit A K, et al. Hybrid Aquila optimizer with arithmetic optimization algorithm for global optimization tasks [J]. Soft Computing, 2022, 26: 4863–4881.

[8] Wang S, Ma J, Li W, et al. An optimal configuration for hybrid SOFC, gas turbine, and Proton Exchange Membrane Electrolyzer using a developed Aquila Optimizer [J]. International Journal of Hydrogen Energy, 2022, 47(14): 8943-8955.

[9] Eric Ofori-Ntow Jnr, Yao Y Z, Rodrigues M J, et al. A hybrid chaotic-based discrete wavelet transform and Aquila optimisation tuned-artificial neural network approach for wind speed prediction [J]. Results in Engineering, 2022, 14: 2590-1230.

[10] Layeb A. Tangent search algorithm for solving optimization problems [J/OL]. Neural Comput & Applic, 2022, 34: 8853–8884. https://doi.org/10.1007/s00521-022-06908-z.

[11] Storn R, Price K. Differential Evolution –A Simple and Efficient Heuristic for global Optimization over Continuous Spaces [J]. Journal of Global Optimization, 1997, 11(4): 341-359.

[12] Shao X, Chen Z, Lin X. Resolution of multicomponent overlapping chromatogram using an immune algorithm and genetic algorithm [J]. Chemometrics & Intelligent Laboratory Systems, 2000, 50(1): 91-99.

[13] Storn R, Price K. Differential Evolution –A Simple and Efficient Heuristic for global Optimization over Continuous Spaces [J]. Journal of Global Optimization, 1997, 11(4): 341-359.

[14] Reynolds A M, Smith A D, Reynolds D R, et al. Honeybees perform optimal scale-free searching flights when attempting to locate a food source [J]. Journal

of Experimental Biology, 2007, 210(21): 3763-3770.

[15] Mantegna R N. Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes [J]. Physical Review, 1994, 49(5): 4677-4683.

[16] Wilcoxon F. Individual comparison by ranking methods [J]. Biometrics, 1945, 1: 80-83.

[17] Li A, Quan L, Cui G, Xie S. Sparrow Search Algorithm Combining Sine-Cosine and Cauchy Mutation [J]. Computer Engineering and Applications, 2022, 58(03): 91-99.

[18] Zhang W, Liu S, Ren C. Mixed Strategy Improved Sparrow Search Algorithm [J]. Computer Engineering and Applications, 2021, 57(24): 74-82.

[19] Liu J, Ma Y, Li Y. Improved whale algorithm for solving engineering design optimization problems [J]. Computer Integrated Manufacturing Systems, 2021, 27(07): 1884-1897.

[20] Coelho L S. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems [J]. Expert Systems With Applications, 2009, 37(2): 1676-1683.

[21] Mirjalili S, Mirjalili S M, Hatamlou A. Multi-Verse Optimizer: a nature-inspired algorithm for global optimization [J]. Neural Computing and Applications, 2016, 27(2): 495-513.

[22] Ma W, Gao Y, Zhao M. Global-best difference-mutation brain storm optimization algorithm [J]. Systems Engineering and Electronics, 2022, 44(1): 270-278.

Note: Figure translations are in progress. See original paper for figures.