U.P.B. Sci. Bull., Series C, Vol. 88, Iss. 3, 2026 ISSN 2286 – 3540

WHALE OPTIMIZATION ALGORITHM BASED ON TENT AND SINE OPTIMIZATION

Jing CHEN [1,*]

To address the limitations of the whale optimization algorithm (WOA), including its tendency to fall into local optima and weak local search capability, an improved version—termed TSVWOA—is proposed in this paper. Specifically, the tent map is adopted for population initialization to enhance population diversity; a sine function-based optimization factor is introduced to balance the local and global optimization performance of individuals; and a hybrid mutation strategy is employed to improve the optimization capability of individuals. In the simulation experiments, TSVWOA is compared with ACO, PSO, WOA, and AWOA on benchmark functions, and its effectiveness is further verified by optimizing cost and time in a network topology model. The results demonstrate that TSVWOA exhibits superior network optimization performance. In terms of average network cost, it significantly outperforms ACO and PSO, achieving reductions of 42.9% and 28.2% compared to WOA and AWOA, respectively. Regarding average network time, it also surpasses ACO and PSO, with reductions of 70.37% and 69.44% compared to WOA and AWOA, respectively. These findings indicate that TSVWOA has favorable potential for practical applications.

Keywords: whale optimization algorithm, sinusoidal function, hybrid variation

1. Introduction

Metaheuristic algorithms are highly effective for solving complex optimization problems. Their core principle involves simulating natural phenomena, biological behaviors, or physical processes to find global or approximate optimal solutions [1]. Common examples include ant colony optimization, particle swarm optimization, and simulated annealing, which are independent of specific problem structures, with strong versatility and flexibility to handle nonlinear, multimodal, and high-dimensional complex optimization problems. In recent years, the whale optimization algorithm (WOA) [2] has attracted great attention due to its ingenious design, ease of implementation, relaxed functional constraints, and minimal parameter tuning requirements. However, it has certain limitations, such as slow convergence, low solution accuracy and susceptibility to local optima. To address these shortcomings, this paper proposes an improved WOA with three key enhancements: tent chaos-based population initialization to improve diversity; integration of a sine function into the whale factor to balance local and global optimization; and mutation strategy in late iteration to enhance individual optimization capability. Simulation experiments using benchmark functions verify

[1] * School of Mechatronics & IT, Yiwu Industrial & Commercial College, Yiwu, 322000, China, Corresponding author: ywcj1981@163.com

its performance, and further validation via network topology model costs confirms its excellent practical effectiveness.

2. Related research

The whale optimization algorithm has attracted extensive attention as a flexible metaheuristic, giving rise to numerous improved variants. Existing studies can be summarized into two streams: algorithmic enhancement and application expansion.

(1) Algorithmic enhancements

Research in this direction focuses on three main aspects. First, in population control and diversity promotion, researchers have improved the initial population structure [3], designed subpopulation division mechanisms [4], and used nondominated sorting to select high-quality solutions [5], thereby enhancing population diversity and exploration capability. Second, in strategy fusion and cross-algorithm integration, external strategies such as water-wave optimization [6], differential evolution [7], artificial immune regulation [8], and particle swarm optimization [9] have been integrated into WOA to strengthen global search and avoid local optima. Third, in parameter and mechanism optimization, adaptive predation strategies [10], multi-vector position updates [11], and multi-strategy combinations [12,13] have been adopted to adjust convergence speed and search stability. However, most of these improvements are designed for specific problems and lack a universal framework for achieving a dynamic balance between exploration and exploitation.

(2) Application expansion

In parameter optimization, WOA has been applied to fuel cell controller parameters [14]. In energy system optimization, improved WOA has been adopted for maximum power point tracking in wind turbines [15]. In feature selection and prediction, hybrid algorithms combining WOA with differential evolution, genetic algorithms, pollination algorithms, or grey wolf optimization have been successfully used in fleet scheduling [16], nonlinear system optimization [17], and logistics vehicle routing [18]. In multi-objective optimization, WOA combined with nondominated sorting achieves balanced solutions for cost and efficiency [19]. In engineering design, a grey wolf-enhanced WOA has been proposed for practical optimization problems [20]. Nevertheless, most improvements remain domain-specific and the theoretical basis of hybrid strategies require further exploration.

In summary, the applications of improved WOA revolve around core demands across various domains: parameter optimization emphasizes accuracy and robustness; resource scheduling prioritizes constraint adaptability; energy system stress disturbance resilience; feature selection and prediction rely on adaptability to data characteristics; hybrid algorithms highlight strategy compatibility; and multi-objective optimization focuses on solution balance.

3. Whale Optimization Algorithm

WOA is a swarm intelligence optimization algorithm proposed by Mirjalili and Lewis. The algorithm takes whales in the ocean as its research subject, incorporates the living habits of whale populations, and divides the algorithm into three stages: encirclement and predation, group attack, and food search. The specific process of the algorithm is detailed in Ref. [2], and we provide a brief description here. In the algorithm, $X_i = (X_i^1, X_i^2, \cdots X_i^D)$, where $i = 1, 2, \cdots, N$ represents the position of the i th whale in a D-dimensional space.

(1) Surrounding predation

In the first phase of the algorithm, the current ideal position is defined as the prey, and all other whales are encircled by the optimal individual, as described in the following mathematical model:

$$X(t+1) = X_p(t) - A \cdot | C \cdot X_p(t) - X(t) | \tag{1}$$

where $X_p = (X_p^1, X_p^2, \cdots X_p^D)$ is the prey location, $A \cdot | C.X_p(t) - X(t) |$ is the bracketing step, t is the current number of iterations, and the mathematical formulae for C and A are given below:

$$C = 2 \cdot rand_2 \tag{2}$$

$$A = 2a \cdot rand_1 - a \tag{3}$$

In the above formulations, $rand_1$ and $rand_2$ represent random integers between [0, 1], respectively, and a is a convergence factor that steadily decreases from 2 to 0. The expressions are given below:

$$a = 2 - 2t / t_{\max} \tag{4}$$

where t and $t_{\max}$ are the current and maximum numbers of iterations, respectively.

(2) Bubble attack

Whales in the ocean use their own spray to create large bubbles to encircle their prey, whereas each individual uses a spiral ascent to update its position. The mathematical model is expressed as follows:

$$X(t+1) = \vec{D} \times e^{lb} \times \cos(2\pi l) + X_p(t) \tag{5}$$

where $\vec{D} = | X_p(t) - X(t) |$ indicates the distance between the whale and its current target, b represents the integer constant that limits the spiral shape, and l represents a random value between -1 and 1.

(3) Food-seeking stage

Each whale randomly picks the places of nearby individuals to undertake a random search, which is mathematically stated as follows:

$$X(t+1) = X_{rand}(t) - A \cdot | C \cdot X_p(t) - X(t) | \tag{6}$$

where X_{rand} is the randomized individual whale position vector.

4. Improved whale optimization algorithm

Like most heuristic algorithms, WOA tends to fall into local optima and performs poorly in high-dimensional problems. To enhance its performance, we propose research in three aspects: population initialization, adaptive factor optimization, and late-stage individual optimization.

(1) Population initialization

For metaheuristic algorithms, population initialization effectiveness directly affects solution accuracy and convergence speed, as has been validated in many applications. We thus use the tent chaos map for population initialization by leveraging the randomness and exploration advantages of chaos algorithms, which helps to uniformly distribute the initial population in the solution space, cover more potential optimal regions, and significantly enhance population diversity. The mathematical expression is as follows:

$$x_{n+1} = \begin{cases} \mu x_n & x_n < \dfrac{1}{2} \\ \mu(1-x_n) & x_n >= \dfrac{1}{2} \end{cases} \tag{7}$$

The new population of individuals is generated by Tent chaos in Eq. (7). Fig. 1 shows the results when the value of μ in the tent map is set to 0.1, 0.5, and 0.9. The results indicate that a more uniform distribution is achieved when μ is 0.1.

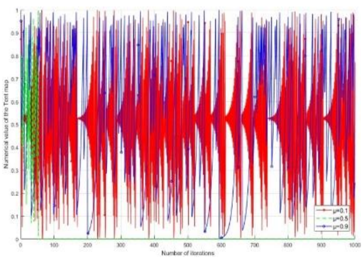


Fig. 1. Different μ values

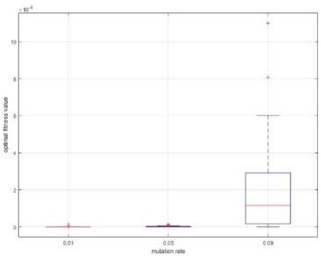


Fig. 2. Boxplot of Sphere function

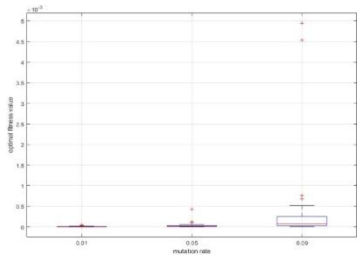


Fig. 3. Boxplot of Schwefel 2.22 function

(2) Convergence factor based on the sine function

In WOA, the convergence factor—controlling search behavior and decreases from 2 to 0—shifts the algorithm from global exploration to local development. To address this, we optimize it using a sine function (with periodicity and nonlinear decrease) to refine its variation: slower early decrease maintains high global exploration intensity to avoid missing the global optimum, while faster late decrease narrows search steps for higher local solution accuracy.

$$a(t) = a_0 \cdot (1+\sin(\frac{\pi t}{2t_{\max}})/2 \tag{8}$$

In Equation (8), a_0 denotes the initial value of the parameter.

(3) Mixed mutation strategy

In later iterations of WOA, most whale individuals concentrate around local optima, losing the ability to explore other potential optimal regions. To address this issue, we propose a hybrid mutation strategy that randomly mutates some dimensions of selected whale individuals with a certain probability to enhance population diversity, helping the algorithm explore more search spaces and increasing the chance of finding the global optimum. Boxplots of the Sphere and Schwefel 2.22 functions (Figs. 2 and 3) show that a 0.01 mutation probability yields an extremely narrow box (high stability), with median and mean close to 0, demonstrating that the algorithm can stably approach the theoretical optimum.

$$x(i)_j = lb + (ub - lb) \times rand(); \; if \; q < 0.01 \tag{9}$$

In Equation (9), *lb* and *ub* denote the lower and upper bounds of the algorithm, respectively.

(4) Algorithm complexity analysis

The time complexity of the TSVWOA is primarily determined by population initialization, fitness evaluation, position updating, and mutation operations. Its key components are as follows: population initialization requires a time complexity of $O(N \times \text{dim})$; in each iteration, both fitness calculation and position updating take a time complexity of $O(N \times \text{dim})$, while the time complexity of the mutation operation is $O(N \times \text{dim} \times q)$. By synthesis, the total time complexity is $O(Max_{iter} \times N \times Dim)$, which is the same order of magnitude as that of the standard WOA. In terms of space complexity, the algorithm needs to store N individuals with dim dimensions, the optimal solution, and historical fitness records. Thus, the space complexity is $O(N \times \text{dim} + Max_{iter})$, which is mainly determined by the population matrix and the iterative record array. Overall, by introducing Tent map initialization and a hybrid mutation strategy, TSVWOA improves convergence performance while maintaining a complexity comparable to that of the standard WOA.

5. Simulation experiments

To evaluate the performance of the proposed TSVWOA, we compared it with ACO, PSO, WOA, and the adaptive whale optimization algorithm (AWOA) [21]. The hardware was selected as Core I7-8700 for the CPU, 16GDDR4 for the RAM, 1TSSD for the hard disk capacity, Win10 for the operating system of the software and MATLAB 2024a for the simulation software. To better illustrate the effectiveness of the TSVWOA algorithm against the comparison algorithms, we conducted experiments of various configurations, employing different population sizes and iteration numbers, all with the ultimate aim of validating the performance of the TSVWOA algorithm

5.1 Algorithm performance comparison

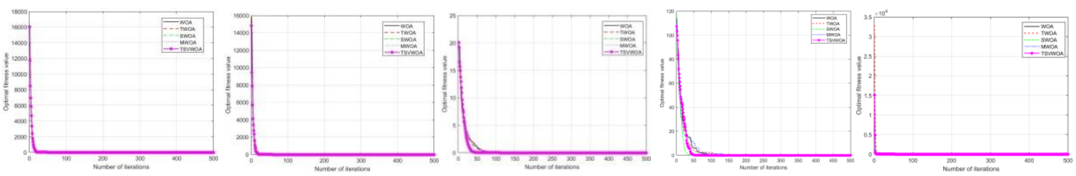
Each algorithm was executed for 30 independent runs on every benchmark function and dimensional setting. The results reported in Table 2 (maximum, minimum, average, and standard deviation) are computed over these 30 runs. We selected five benchmark functions (Sphere, Step, Ackley, Rastrigin, and Schwefel 2.22) to test the performance of the five algorithms, using four evaluation indicators: maximum (worst-case), minimum (optimal potential), average (overall effectiveness), and standard deviation (result stability). These four metrics comprehensively assess the algorithm's accuracy, stability, and overall performance, avoiding single-indicator limitations. The relevant parameter settings for the five algorithms are listed in Table 1 (with 100 iterations), and their benchmark function results are shown in Table 2.

Table 1.

Main parameters of the five algorithms.

Algorithm	Description of the main parameters
ACO	Number of ants: 50; pheromone evaporation rate: 0.5; pheromone factor: 1; heuristic factor: 2; probability: 0.9.
PSO	Number of particles: 50; inertia weight: 0.7; cognitive and social variables of 1.4 each.
WOA	Number of whales: 50; parameter *a*: 2; probability: 0.5.
AWOA	Number of whales: 50; parameter *a*: 2; probability: 0.5.
TSVWOA	Number of whales: 50; parameter a: 2; probability: 0.5; Tent map parameter μ: 0.1 (selected through empirical testing, see Fig. 1); mutation probability q: 0.01 (selected through empirical testing, see Figs. 2 and 3.

In the ablation experiment, we compared TWOA (WOA with only tent map initialization), SWOA (WOA with only sine convergence factor), and MWOA (WOA with only mutation) with the original WOA and our proposed TSVWOA on five benchmark functions (population size = 30, iterations = 500, dimension = 10, 10 independent runs). The ablation study uses a smaller population size (30 instead of 50) and more iterations (500 instead of 100) in order to isolate each component's individual contribution under extended search conditions, thereby making convergence differences between variants more visible. The results are shown in Fig. 4. From the convergence curves, TSVWOA shows significant performance superiority across different optimization problems.



(a) Sphere (b) Step (c) Ackley (d) Rastrigin (e) Schwefel 2.22

Fig. 4. Ablation experiments on five algorithms

Whether for high initial fitness value scenarios (e.g., Sphere, Step functions) or low initial value, narrow range tasks (Ackley, Rastrigin, Schwefel 2.22 functions), TSVWOA converges much faster than WOA, TWOA, SWOA, and MWOA, quickly dropping to near optimal fitness value within few iterations and remaining stable. Unlike other algorithms with slow initial convergence, late-stage fluctuations or stagnation, TSVWOA has clear advantages in convergence efficiency and maintains better final optimization accuracy, confirming its effectiveness in optimization ability and convergence characteristics.

Table 2.

Test results of the five algorithms on the benchmark function.

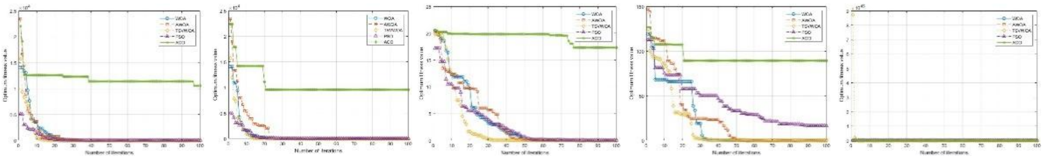
Function	Algorithm	Dimension	Max value	Min value	Average	Std	95% confidence interval
Step	ACO	2	1.600000e+01	0.000000e+00	5.000000e+00	5.312459e+00	[3.09, 6.91]
		10	1.442300e+04	7.364000e+03	1.084980e+04	2.102713e+03	[10057.6, 11642.0]
		30	6.627700e+04	4.834500e+04	5.980300e+04	5.943404e+03	[57667.2, 61938.8]
	PSO	2	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	[0.00, 0.00]
		10	1.000000E+00	0.000000E+00	2.000000E-01	4.216370E-01	[-0.09, 0.49]
		30	1.959000E+03	6.670000E+02	1.116200E+03	3.497770E+02	[999.5, 1232.9]
	WOA	2	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	[0.00, 0.00]
		10	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	[0.00, 0.00]
		30	2.000000E+00	0.000000E+00	2.000000E-01	6.324555E-01	[-0.03, 0.43]
	AWOA	2	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	[0.00, 0.00]
		10	1.000000E+00	0.000000E+00	1.000000E-01	3.162278E-01	[-0.08, 0.28]
		30	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	[0.00, 0.00]
	TSVWOA	2	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	[0.00, 0.00]
		10	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	[0.00, 0.00]
		30	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	[0.00, 0.00]
Sphere	ACO	2	1.001171 E +01	1.578994 E -01	4.117147e+00	3.347166 E +00	[2.08, 6.15]
		10	1.245695 E +04	3.818620 E +03	7.614529e+03	2.446733 E +03	[6582.3, 8646.8]
		30	6.454555 E +04	5.040951 E +04	5.698561e+04	4.451827 E +03	[55178.1, 58793.1]
	PSO	2	4.118494 E -11	6.755655 E -14	7.211037e-12	1.218752 E -11	[-1.91e-12, 1.63e-11]
		10	5.256290E-04	1.958658E-05	1.143179E-04	1.546910E-04	[-1.31e-05, 2.42e-04]
		30	1.636678 E +01	2.745257 E +00	7.954165e+00	4.949333 E +00	[5.92, 9.99]
	WOA	2	7.627591E-18	3.710190E-29	7.627959E-19	2.412043E-18	[-2.11e-19, 1.74e-18]
		10	2.974381E-08	9.457970E-14	6.135643E-09	1.050778E-08	[-2.32e-09, 1.46e-08]
		30	4.166488E-08	3.934422E-11	7.148050E-09	1.370249E-08	[-3.02e-09, 1.73e-08]
	AWOA	2	1.764301E-21	5.699190E-29	2.083247E-22	5.532558E-22	[-2.23e-22, 6.39e-22]
		10	9.415402E-08	1.599313E-12	1.561106E-08	2.933103E-08	[-3.07e-09, 3.43e-08]
		30	1.009744E-06	5.338355E-13	1.095734E-07	3.168715E-07	[-1.02e-07, 3.21e-07]
	TSVWOA	2	1.326978E-24	8.276370E-34	1.332493E-25	4.194371E-25	[-1.88e-25, 4.55e-25]
		10	1.388721E-12	3.164262E-18	2.326708E-13	4.470119E-13	[-1.18e-13, 5.83e-13]
		30	1.912409E-11	8.778822E-17	3.006049E-12	5.891430E-12	[-1.02e-12, 7.03e-12]
Ackley	ACO	2	4.557300E+00	1.777867E+00	3.513181E+00	9.199600E-01	[3.17, 3.86]
		10	1.959173E+01	1.807230E+01	1.872359E+01	4.760905E-01	[18.55, 18.89]
		30	2.067838E+01	2.007273E+01	2.040794E+01	1.803228E-01	[20.34, 20.47]
	PSO	2	7.452055E-06	3.982349E-07	2.236893E-06	2.115794E-06	[1.41e-06, 3.06e-06]
		10	2.513240E-02	6.299094E-03	1.139906E-02	5.971740E-03	[9.15e-03, 1.36e-02]
		30	5.364858E+00	3.414645E+00	4.393265E+00	7.149708E-01	[4.11, 4.68]
	WOA	2	1.660235E-07	2.124967E-12	1.905802E-08	5.183593E-08	[-2.31e-08, 6.12e-08]
		10	8.024367E-04	3.758111E-06	2.465619E-04	2.628398E-04	[1.33e-04, 3.60e-04]
		30	8.510105E-04	1.678327E-05	2.442567E-04	2.548592E-04	[1.34e-04, 3.54e-04]
	AWOA	2	1.087860E-07	1.475131E-11	1.219999E-08	3.405216E-08	[-8.33e-09, 3.27e-08]
		10	2.292415E-04	5.489488E-06	7.067323E-05	8.374370E-05	[3.80e-05, 1.03e-04]
		30	7.384138E-04	2.875811E-06	1.619912E-04	2.255966E-04	[7.32e-05, 2.51e-04]
	TSVWOA	2	5.155876E-13	4.440892E-16	7.291945E-14	1.600098E-13	[-1.00e-14, 1.56e-13]
		10	6.734328E-07	1.114506E-08	2.442889E-07	2.096217E-07	[1.65e-07, 3.24e-07]
		30	3.280850E-07	3.644967E-09	8.051486E-08	1.072590E-07	[3.97e-08, 1.21e-07]
Rastrigin	ACO	2	1.180090E+00	7.891558E-02	4.766118E-01	3.648112E-01	[3.05, 6.48]
		10	8.599657E+01	5.590242E+01	7.389057E+01	9.081387E+00	[69.87, 77.91]

		30	3.865549E+02	3.539048E+02	3.724010E+02	1.077335E+01	[367.94, 376.86]
		2	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	[0.00, 0.00]
	PSO	10	3.382853E+01	2.984877E+00	1.472536E+01	1.031645E+01	[10.45, 18.99]
		30	1.354561E+02	6.765709E+01	1.121811E+02	1.911736E+01	[113.28, 129.16]
		2	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	[0.00, 0.00]
	WOA	10	1.421085E-14	0.000000E+00	1.421085E-15	4.493867E-15	[-2.11e-15, 4.95e-15]
		30	5.684342E-14	0.000000E+00	1.136868E-14	2.396729E-14	[-1.33e-14, 3.60e-14]
		2	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	[0.00, 0.00]
	AWOA	10	3.718916E+01	0.000000E+00	3.718916E+00	1.176024E+01	[-4.03, 11.47]
		30	5.684342E-14	0.000000E+00	1.136868E-14	2.396729E-14	[-1.33e-14, 3.60e-14]
		2	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	[0.00, 0.00]
	TSVWOA	10	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	[0.00, 0.00]
		30	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	[0.00, 0.00]
		2	2.675479E+00	4.654986E-01	1.331580E+00	6.561095E-01	[1.01, 1.65]
	ACO	10	6.208855E+09	2.422439E+06	1.657631E+09	2.140517E+09	[-3.02e+08, 3.62e+09]
		30	9.632064E+37	1.326992E+36	1.667745E+37	2.897966E+37	[-1.98e+37, 5.32e+37]
		2	1.012576E-26	3.549807E-30	1.591733E-27	3.151947E-27	[-4.02e-28, 3.59e-27]
	PSO	10	3.000000E+02	4.371870E-16	6.000003E+01	9.660916E+01	[-28.24, 148.24]
		30	8.711830E+02	1.323474E+00	3.177954E+02	2.369725E+02	[223.8, 411.79]
		2	2.128159E-41	4.227747E-55	2.128223E-42	6.729807E-42	[-9.41e-43, 5.20e-42]
Schwefel 2.22	WOA	10	3.007210E-24	3.792476E-28	6.452018E-25	1.108741E-24	[-3.21e-25, 1.61e-24]
		30	9.019434E-20	3.921003E-27	9.082791E-21	2.850032E-20	[-4.18e-20, 2.23e-19]
		2	1.685168E-39	9.317024E-53	1.686308E-40	5.328570E-40	[-3.12e-40, 6.49e-40]
	AWOA	10	6.176137E-22	1.520670E-30	6.250612E-23	1.950492E-22	[-1.03e-22, 2.28e-22]
		30	5.541863E-22	1.945456E-26	8.567268E-23	1.842353E-22	[-9.23e-23, 2.64e-22]
		2	6.001718E-38	1.734481E-42	8.341884E-39	1.845359E-38	[-9.32e-39, 2.60e-38]
	TSVWOA	10	1.838992E-31	2.231714E-38	3.770388E-32	6.120905E-32	[-1.98e-32, 9.52e-32]
		30	1.147571E-30	1.063439E-34	2.136075E-31	3.495637E-31	[-9.82e-32, 5.25e-31]

Table 2 compares the performance of ACO, PSO, WOA, AWOA, and TSVWOA on the five benchmark functions (Step, Sphere, Ackley, Rastrigin, Schwefel 2.22) under 2D, 10D, and 30D tests. TSVWOA significantly outperforms the other algorithms. For the Step and Rastrigin functions, its maximum, minimum, and mean values stably converge to 0 across all dimensions (no standard deviation or 95% confidence interval fluctuations, achieving precise optimization). It is particularly worth pointing out that TSVWOA and WOA reach machine-precision on Rastrigin, and that numerical differences at this scale are not practically meaningful. For the Sphere, Ackley, and Schwefel functions, its mean values are much lower than those of the comparison algorithms (e.g., 3.00e-12 on 30D Sphere, far lower than ACO's 1.67e+37 on 30D Schwefel), with extremely narrow standard deviation and confidence interval. Overall, TSVWOA maintains high optimization accuracy and remarkable stability across multidimensional and multitype benchmarks, effectively overcoming the large errors and fluctuations exhibited by the comparison algorithms in high-dimensional scenarios.

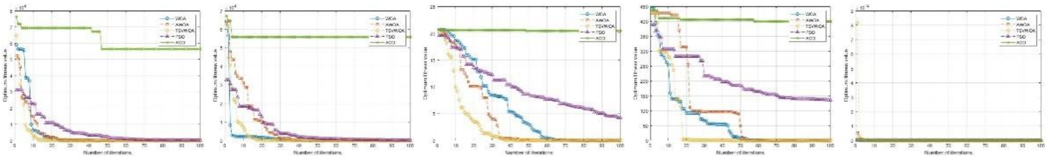
Figs. 5 and 6 show the iteration number and optimal fitness of the five algorithms under 10D and 30D conditions. For the Step function, all algorithms decrease with iterations; except for ACO, the others converge near 0. TSVWOA converges fastest under 30D and exhibits better stability as dimensionality increases. For the Sphere function, TSVWOA achieves the best fitness with smaller fluctuations and fewer iterations under both 10D and 30D. For the Ackley function, TSVWOA performs moderately well under 10D, while its convergence advantage becomes significant under 30D with higher accuracy and better stability. For the Rastrigin function, TSVWOA converges rapidly and stably under both dimensions, far outperforming the other algorithms. For the Schwefel 2.22 function, all algorithms perform similarly, but TSVWOA still shows a slight speed advantage.

In summary, TSVWOA achieves better fitness, faster convergence, and higher stability under high-dimensional conditions, showing strong potential for large-scale optimization problems.



(a) Sphere (b) Step (c) Ackley (d) Rastrigin (e) Schwefel 2.22

Fig. 5. Comparison of the five algorithms in terms of 10 Dimension



(a) Sphere (b) Step (c) Ackley (d) Rastrigin (e) Schwefel 2.22

Fig. 6. Comparison of the five algorithms in terms of 30 Dimension

Table 3.

Significance levels of the Wilcoxon rank-sum test

Function	Algorithm vs TSVWOA	*P* value	Significant (α = 0.05)
Step	ACO vs TSVWOA	0.000002	1
	PSO vs TSVWOA	0.000002	1
	WOA vs TSVWOA	1.000000	0
	AWOA vs TSVWOA	1.000000	0
Sphere	ACO vs TSVWOA	0.000002	1
	PSO vs TSVWOA	0.000002	1
	WOA vs TSVWOA	0.001114	1
	AWOA vs TSVWOA	0.000388	1
Ackley	ACO vs TSVWOA	0.000002	1
	PSO vs TSVWOA	0.000002	1
	WOA vs TSVWOA	0.001593	1
	AWOA vs TSVWOA	0.001484	1
Rastrigin	ACO vs TSVWOA	0.000002	1
	PSO vs TSVWOA	0.000002	1
	WOA vs TSVWOA	0.000963	1
	AWOA vs TSVWOA	0.009842	1
Schwefel 2.22	ACO vs TSVWOA	0.000002	1
	PSO vs TSVWOA	0.000002	1
	WOA vs TSVWOA	0.000006	1
	AWOA vs TSVWOA	0.000261	1

Table 3 presents the Wilcoxon rank-sum test results (α = 0.05, 30 independent runs, sample size is 30) comparing TSVWOA with five algorithms (population size = 50, iterations = 100). TSVWOA significantly outperforms ACO and PSO across all functions, with all p-values equal to 0.000002 (well below 0.05). When compared with WOA and AWOA, results differ slightly: on the Step function, the p-values equal 1.000, indicating no significant difference. However, on Sphere, Ackley, Rastrigin, and Schwefel 2.22, all p-values are below 0.05, confirming TSVWOA's significant superiority. Overall, TSVWOA is competitive, outperforming ACO and PSO, and showing significant advantages over WOA and AWOA on four of the five benchmarks.

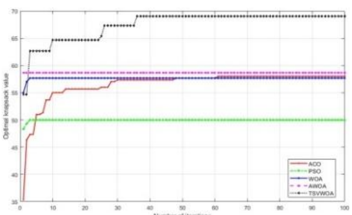


Fig. 7. Discrete optimization

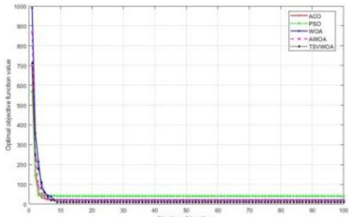


Fig. 8. g01 function

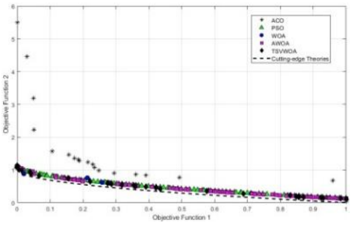


Fig. 9. ZDT1 function

Each algorithm was executed for 30 independent runs in the following experiments. Figs. 7–9 compare TSVWOA with four other algorithms (population size = 30, iterations = 100) in discrete, constrained, and multi-objective optimizationIn discrete optimization, TSVWOA's convergence curve rapidly reaches high-quality solutions early and maintains optimal levels stably, outperforming the other algorithms in both convergence speed and final optimization results, thereby verifying its efficiency and capability in this scenario.

In the constrained optimization of the g01 function, TSVWOA quickly converges to a better objective value and remains stable initially, while the other algorithms lag in convergence and have lower final accuracy, indicating TSVWOA's dual advantages in speed and accuracy.

In the multi-objective optimization of the ZDT1 function, TSVWOA's solution set (black dots) better conforms to the Pareto front with uniform, dense distribution, outperforming the other algorithms whose solution sets deviate or are sparse, demonstrating its ability to generate high-quality, diverse solutions.

5.2. Scenario application

To further demonstrate TSVWOA's practical effectiveness, we construct a network topology optimization model for real communication networks. Based on node functional classification, this model establishes a multi-constraint, multi-objective optimization framework, that achieves coordinated optimization of network cost, transmission delay, and operational reliability by quantifying the differential characteristics of links and nodes.

The model divides the network into three functional layers (core, convergence, access), assigning differentiated attributes to each node according to its layer to simulate practical network functional division. Link performance parameters (capacity, cost, delay, reliability) depend on the hierarchical and physical characteristics of the nodes at both ends. The model aims to minimize the weighted sum of cost, delay, and reliability objectives for multi-objective coordinated optimization. Definitions of the main symbols used are shown in Table 4.

Table 4.

Definitions of the main symbols

Symbol	Description	Symbol	Description
n	Total number of network nodes	R_{ij}	Link reliability between node i and node j
L	Number of network layers	W_i	Weight of node i
$layer_i$	Hierarchy of node i ; 1–3 represent the core layer, convergence layer, and access layer in sequence	B_i	Bandwidth of node i
x_{ij}	Connection state between node i and node j ; 1 denotes connected, 0 denotes disconnected	Cap_{ij}	Link capacity between node i and node j
C_{ij}	Link cost between node i and node j	$D_i^{\max}$	Maximum number of connections of node i
T_{ij}	Link delay between node i and node j	α , β , γ	Weight coefficients of cost, delay and reliability

Some of the parameters in the experiment are as follows: the number of nodes is 15, the number of network layers is 3, the cost weight is 0.5, the latency weight is 0.3, and the reliability weight is 0.2. The bandwidth requirement for core layer nodes is 100 Mbps, with a weight of 0.8 and a maximum number of connections of 10; the maximum link capacity is 2000 Gbps. The bandwidth requirement for aggregation layer nodes is 60 Mbps, with a weight of 0.5 and a maximum number of connections of 6, and the maximum link capacity is 1000 Gbps. The bandwidth requirement for access layer nodes is 30 Mbps, with a weight of 0.2 and a maximum number of connections of 2, and the maximum link capacity is 200 Gbps.

(1) Total transmission cost

The total connection cost mainly considers the cost of each link and its coupling relationship with the importance weight of nodes. Its specific expression is as follows:

$$\mathrm{Cos}t = \sum_{i=1}^{n}\sum_{j=i+1}^{n} x_{ij} \cdot C_{ij} \cdot W_i \cdot W_j \tag{10}$$

The link cost is calculated as follows: $C_{ij} = Base\mathrm{Cos}t_{ij} + Length_{ij} \cdot 0.1 + Cap_{ij} \cdot 0.001$, $Base\mathrm{Cos}t_{ij} \in [1,10]$, and $Length_{ij} \in [1,50]$.

(2) Total transmission delay time

The total transmission delay refers mainly to the sum of the delays of all links. Its specific expression is given as:

$$Time = \sum_{i=1}^{n}\sum_{j=i+1}^{n} x_{ij} \cdot T_{ij} \tag{11}$$

The link delay T_{ij} is calculated as follows:

$$T_{ij} = \left(\frac{Length_{ij}}{2\times10^5}\right)\times1000 + (0.001+0.009\times rand())\cdot(layer_i + layer_j) \tag{177 2)}$$

where $\frac{Length_{ij}}{2\times10^5}$ denotes the delay propagation, $0.001+0.009\times rand()$ denotes the processing delay, and $layer_i$ and $layer_j$ represent the i -th and j -th core layers, respectively.

(3) Reliability penalty

Link reliability mainly refers to the probability that a link operates normally within a unit time. The relevant links in the core layer have higher reliability than those in the edge layer because of their superior hardware specifications and more frequent maintenance.

$$\mathrm{Re}liability\_Penalty = \sum_{i=1}^{n}\sum_{j=i+1}^{n} x_{ij}\cdot(1-R_{ij})\times100 \tag{13}$$

(4) Constraints

To ensure that the optimized network topology meets the requirements of practical operation, the model defines the following five types of constraints:

Connectivity Constraint: The network topology must be a simply connected graph, with a path existing for node $\forall i, j \in [1,n]$.

Bandwidth Constraint: $\sum_{j=1}^{n} x_{ij}\cdot Cap_{ij} \ge B_i, \forall i \in [1,n]$

Connection Number Constraint: $1 \le \sum_{j=1}^{n} x_{ij} \le D_i^{\max}, \forall i \in [1,n]$

Hierarchical connection constraint:

$$\sum_{k\in \text{Core layer node}} x_{ik} \ge 1, \forall i \in \text{Convergence layer node}$$

(5) Optimization Objective Function

The objective of the optimization of this network model is the weighted processing of cost, delay, and reliability to achieve multi-objective coordinated optimization. Therefore, the network objective function is expressed as:

$$\min F = \alpha\cdot \mathrm{Cos}t + \beta\cdot Time + \gamma\cdot \mathrm{Re}l\_pen \tag{14}$$

where α , β and γ are the weights of the three indicators, and their sum equals 1.

Each algorithm was executed for 30 independent runs in the network topology experiment. Figs. 10–12 compare the five algorithms (population size = 30, iterations = 100) in terms of objective function value, total cost, and time consumption. Fig. 10 shows all algorithms' objective function values decrease with iterations, with TSVWOA having a distinct advantage. Fig. 11 indicates TSVWOA outperforms ACO and PSO significantly; its total cost is 42.9% lower than that of WOA and 28.2% lower than that of AWOA. Fig. 12 shows that TSVWOA is superior to ACO and PSO in time consumption, which is 70.37% and 69.44% lower than WOA and AWOA, respectively. Overall, the above results demonstrate that TSVWOA can effectively reduce the cost and time consumption of the network.

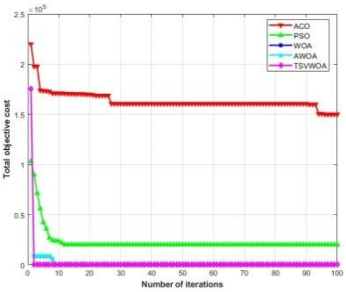


Fig. 10. Objective Function Values

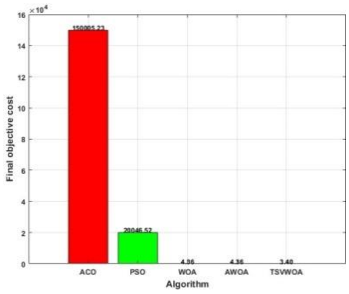


Fig. 11. Cost comparison

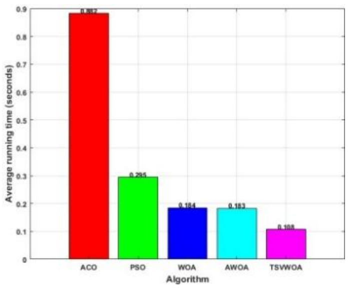


Fig. 12. Time comparison

6. Conclusion

Reducing the cost and connection time of nodes in the network topology is key to enhancing network performance. We have proposed TSVWOA to address this problem. However, the current model mainly considers cost and connection time, without accounting for the complexity of the network model or the diversity of nodes in the network equipment and so on. In future work, we will validate the algorithm using more complex network topology models while continuously improving its performance.

Acknowledgment

The research is supported by the General Project of Zhejiang Province Philosophy and Social Science Planning Project (22NDJC194YB).

R E F E R E N C E S

[1] Dokeroglu, T., Sevinc, E., Kucukyilmaz, T., & Cosar, A. (2019). A survey on new generation metaheuristic algorithms. Computers & Industrial Engineering, 137, 106040.
[2] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in engineering software, 95, 51-67.
[3] Abd Elaziz, M., & Mirjalili, S. (2019). A hyper-heuristic for improving the initial population of whale optimization algorithm. Knowledge-Based Systems, 172, 42-63.
[4] Shen, Y., Zhang, C., Gharehchopogh, F. S., & Mirjalili, S. (2023). An improved whale optimization algorithm based on multi-population evolution for global optimization and engineering design problems. Expert Systems with Applications, 215, 119269.

[5] Jangir, P., & Jangir, N. (2017). Non-dominated sorting whale optimization algorithm (NSWOA): a multi-objective optimization algorithm for solving engineering design problems. Glob. J. Res. Eng, 17, 15-42.
[6] Singh, H., Rai, V., Kumar, N., Dadheech, P., Kotecha, K., Selvachandran, G., & Abraham, A. (2023). An enhanced whale optimization algorithm for clustering. Multimedia tools and applications, 82(3), 4599-4618.
[7] Rana, N., Abd Latiff, M. S., Abdulhamid, S. I. M., & Misra, S. (2022). A hybrid whale optimization algorithm with differential evolution optimization for multi-objective virtual machine scheduling in cloud computing. Engineering Optimization, 54(12), 1999-2016.
[8] Sun, L., Wang, X., Ding, W., & Xu, J. (2022). TSFNFR: Two-stage fuzzy neighborhood-based feature reduction with binary whale optimization algorithm for imbalanced data classification. Knowledge-Based Systems, 256, 109849.
[9] Uzer, M. S., & Inan, O. (2023). Application of improved hybrid whale optimization algorithm to optimization problems. Neural Computing and Applications, 35(17), 12433-12451.
[10] Pham, V. H. S., Nguyen, V. N., & Nguyen Dang, N. T. (2024). Hybrid whale optimization algorithm for enhanced routing of limited capacity vehicles in supply chain management. Scientific Reports, 14(1), 793.
[11] Nadimi-Shahraki, M. H., Farhanginasab, H., Taghian, S., Sadiq, A. S., & Mirjalili, S. (2024). Multi-trial vector-based whale optimization algorithm. Journal of Bionic Engineering, 21(3), 1465-1495.
[12] Li, M., Yu, X., Fu, B., & Xu, W. (2025). A modified whale optimization algorithm with multi-strategy mechanism for global optimization problems. Neural Computing and Applications, 37(27), 22339-22352.
[13] Qu, S. Z., Liu, H., Xu, Y., Wang, L., Liu, Y., Zhang, L., Song, J., & Li, Z. (2024). Application of spiral enhanced whale optimization algorithm in solving optimization problems. Scientific Reports, 14(1), 24534.
[14] Percin, H. B., & Caliskan, A. (2023). Whale optimization algorithm based MPPT control of a fuel cell system. International journal of hydrogen energy, 48(60), 23230-23241.
[15] Qais, M. H., Hasanien, H. M., & Alghuwainem, S. (2020). Enhanced whale optimization algorithm for maximum power point tracking of variable-speed wind generators. Applied soft computing, 86, 105937.
[16] Singh, A., & Khamparia, A. (2020). A hybrid whale optimization-differential evolution and genetic algorithm based approach to solve unit commitment scheduling problem: WODEGA. Sustainable computing: informatics and systems, 28, 100442.
[17] Tawhid, M. A., & Ibrahim, A. M. (2021). Solving nonlinear systems and unconstrained optimization problems by hybridizing whale optimization algorithm and flower pollination algorithm. Mathematics and Computers in Simulation, 190, 1342-1369.
[18] Pham, V. H. S., Nguyen, V. N., & Nguyen Dang, N. T. (2024). Hybrid whale optimization algorithm for enhanced routing of limited capacity vehicles in supply chain management. Scientific Reports, 14(1), 793.
[19] Abdel-Basset, M., Mohamed, R., & Mirjalili, S. (2021). A novel whale optimization algorithm integrated with Nelder–Mead simplex for multi-objective optimization problems. Knowledge-Based Systems, 212, 106619.
[20] Gu, Y., Wei, J., Li, Z., Lu, B., Pan, S. & Cheong, N. (2025). GWOA: A multi-strategy enhanced whale optimization algorithm for engineering design optimization. Plos one, 20(9), e0322494.
[21] Cao, D., Xu, Y., Yang, Z., Dong, H., & Li, X. (2023). An enhanced whale optimization algorithm with improved dynamic opposite learning and adaptive inertia weight strategy. Complex & Intelligent Systems, 9(1), 767-795.