



A NOVEL APPROACH: THREE-GROUP EXPLORATION STRATEGY ALGORITHM FOR SOLVING OPTIMIZATION PROBLEMS

*Corresponding author
ayad.ali@uoz.edu.krd

Ayad Ramadhan Ali*

Department of Mathematics, College of Science, University
of Zakho, Zakho, Kurdistan Region, Iraq

Article history:

Received January 2, 2025

Revised June 25, 2025

Accepted July 5, 2025

Keywords:

Swarm optimization;

Group dynamics;

Group Exploration Strategy
Algorithm;

Benchmark test functions;
convergence speed.

Abstract

In this study, we present a novel optimization technique, known as the Three-Group Exploration Strategy (TGES) algorithm, specifically inspired by collaborative group dynamics often seen in problem-solving. We showed wide testing on 26 widely-recognized benchmark functions, providing a severe comparison between TGES and several well-established optimization algorithms. These results highlight TGES's effectiveness in finding optimal solutions with high reliability and accuracy. Furthermore, the practical applications of TGES are demonstrated by successfully solving six interesting, real-world engineering problems, showcasing its adaptability and robustness. The experimental results indicate that TGES not only exhibits superior optimization performance, but it also achieves faster convergence and higher solution quality compared to several leading algorithms. This finds TGES algorithm as a strong and adaptable tool for solving a variety of engineering optimization problems.

1.0 INTRODUCTION

The optimization problems are common in engineering applications such as knapsack problems, data clustering, data classification, path planning, robot control, and so on. It is well known that the swarm intelligence (SI) optimization algorithms have been used as primary techniques to solve global optimization problems because of its simplicity, flexibility and high efficiency. It should be mentioned that, the SI optimization algorithms mainly introduce the randomness in the search process, which is different from deterministic approaches. Note that the deterministic algorithm is easy to get trapped in local optimal solutions in the complex situation. Therefore, it is of practical importance to employ the SI optimization algorithm so as to obtain an optimal solution to the global optimization problem [1].

In recent years, many new swarm intelligence optimization algorithms have been proposed, such as Ant Colony Optimization (ACO) [2], Particle Swarm Optimization (PSO) [3][4], Cuckoo Search (CS) [5], Grey Wolf Optimization (GWO) [6], Whale Optimization Algorithm (WOA) [7], Tunicate Swarm Algorithm (TSA) [8], Sparrow Search Algorithm (SSA) [1] and so on. These algorithms are valued for their high adaptability, minimal parameter requirements, and strong global search capabilities, making them highly effective in addressing practical engineering design optimization problems.

As engineering technology advances, complex engineering problems have become more frequent, prompting the need to improve the diversity and performance of swarm intelligence algorithms. The

key to enhancing algorithm performance lies in achieving an effective balance between exploration (searching through diverse areas of the solution space) and exploitation (refining the search in promising areas) [9]. Both aspects are crucial: exploration helps investigate new regions, while exploitation focuses on refining the best solutions. Striking a balance between the two is essential for improving algorithmic performance.

On the other hand, we found that although each algorithm has its advantages, there are also shortcomings by a deeper investigation. For instance, the ACO algorithm has the disadvantage of slow search speed, and the PSO algorithm has the disadvantage of easy premature convergence. Therefore, it is very important to enhance the current optimization algorithm. According to the no-free-lunch (NFL) theorem [10], the expected performance of each algorithm is the same for solving all optimization problems. In other words, an optimization algorithm may perform well in a series of problems and show poor performance in a different series of problems. Obviously, we can solve the different problems by proposing new optimization algorithms. At the same time, the newly proposed optimization algorithm provides a new solution to solve a complex global optimization problem.

In summary, the structure of this paper is as follows: Section 2 introduces the Three-Group Exploration Strategy (TGES) algorithm, which is inspired by group collaboration tactics. Section 3 presents comparative tests of the TGES algorithm using benchmark test functions. In Section 4, the TGES algorithm is successfully applied to six engineering problems. Finally, Section 5 concludes the paper.

2.0 THREE-GROUP EXPLORATION STRATEGY ALGORITHM (TGES)

In this section we will introduce the Three-Group Exploration Strategy (TGES) technique. The Three-Group Exploration Strategy (TGES) algorithm is a novel population-based optimization approach and its idea is that divides the search agents into three groups, each with a distinct update strategy to explore the search space effectively. The algorithm depends on three types of movements to balance exploration and exploitation throughout the optimization process.

2.1 Mathematical model and algorithm

First, we need to perform a series of initialization preparations. We set the number of particles to be n , the dimension of the problem to be optimized to be d , and use a matrix to simulate the position of the particles X :

$$X = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1d} \\ X_{21} & X_{22} & \dots & X_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ X_{n1} & X_{n2} & \dots & X_{nd} \end{bmatrix} \quad (1)$$

And use $f([X_{n1}, X_{n2}, \dots, X_{nd}])$ to represent the fitness value of the n th individual, the fitness value of the population can be represented by the following vector:

$$F = \begin{bmatrix} f([X_{11}, X_{12}, \dots, X_{1d}]) \\ f([X_{21}, X_{22}, \dots, X_{2d}]) \\ \vdots \\ f([X_{n1}, X_{n2}, \dots, X_{nd}]) \end{bmatrix} \quad (2)$$

At each iteration t , the population is divided into three groups (group 1, group 2 and group 3) and the particles of each group are selected randomly from the population using a non-repeating random permutation. This ensures that each particle is assigned to exactly one group at each iteration. This random shuffling enhances diversity and avoids premature convergence. The goal is to minimize the objective function $f(X)$. The boundary constraints are defined by lb (lower bounds) and ub (upper bounds), ensuring that all solutions stay within a feasible region.

The three groups are updated using different strategies as follows:

The particle's position in group 1 is updated using an oscillatory movement inspired by trigonometric functions. The position of each particle X_i in this group is updated using:

$$X_i^{t+1} = A_t \cos(\omega t + \theta) X_i^t \cdot (1 - X_i^t), \quad (3)$$

Where, X_i^t is the solution in i th position at iteration t , $A_t = A_0 - t(\frac{A_0}{G})$ is linearly decrease amplitude from the initial amplitude A_0 , G is the maximum number of iterations, ω is the angular frequency and θ is the phase constant. The oscillatory nature of the update encourages exploration of the search space while continuing some proximity to the current best solution.

After individual positions in group 1 were updated, an adaptive quadratic movement has been used to update group 2 particles position, which leverages a random parameter r_1 to control the step size and direction:

$$X_i^{t+1} = \begin{cases} (1 - r_1)^2 X_i^t + 2(1 - r_1) r_1 (X_{best}^t - X_i^t), & \text{if } r_2 \leq 0.5 \\ (1 - r_1)^2 X_i^t - 2(1 - r_1) r_1 (X_{best}^t - X_i^t), & \text{if } r_2 > 0.5 \end{cases} \quad (4)$$

Where, X_{best}^t is the best solution at iteration t . r_1 and r_2 are uniformly distributed random numbers in the range $[0,1]$. This movement allows group 2 particles to either move toward or away from the best solution, certifying a balance between exploration and exploitation.

For updating group 3 particles position uses both random perturbations and noise to refine solutions. The update rule for group 3 is:

$$X_i^{t+1} = \begin{cases} X_{best}^t + a_t \mathfrak{N}(0, I) X_i^t, & \text{if } r_3 < 0.8 \\ X_i^t + a_t M, & \text{if } r_3 > 0.8 \end{cases} \quad (5)$$

Where, $a_t = 1 - \frac{t}{G}$ is a linearly decreasing from 1 to 0, $\mathfrak{N}(0, I)$ is Gaussian noise with zero mean, r_3 is uniformly distributed random numbers in the range $[0,1]$, and M is a vector of random binary values (0 or 1). This strategy ensures that individuals in group 3 perform a more stochastic exploration of the search space, allowing the algorithm to escape local optima.

After position of each particle in all groups were updated the fitness of the updated solutions is evaluated using the objective function $f(X)$. The replacement strategy applied to ensures that only better solutions are taken. Specifically, for each particle i , if the fitness of the new solution X_i^{t+1} is better than or equal to the current solution X_i^t , the particle's position is updated; otherwise, it holds the old position:

$$X_i^{t+1} = \begin{cases} X_i^t, & \text{if } f(X_i^t) < f(X_i^{t+1}) \\ X_i^{t+1}, & \text{otherwise} \end{cases} \quad (6)$$

Finally, the convergence of the algorithm is tracked through the best fitness value, which is recorded at each iteration. The algorithm terminates when the maximum number of iterations is reached.

Algorithm 1: The framework of the TGES algorithm

Input:

n : Population size, D : Dimension of the problem, G : Maximum number of iterations, LB : Lower bound of the search space, UB : Upper bound of the search space, $Group1$: number of group1 particles, $Group2$: number of group2 particles, $Group3$: number of group2 particles, A_0 : initial amplitude, ω : angular frequency and θ : phase constant.

Output:

The optimal solution X_{best} and the optimal fitness value f_{best} .

Start:

Initialize the population of n particles randomly within $[LB, UB]^D$ and evaluate the fitness of each particle.

Rank the fitness values and find the current best individual X_{best} .

Set the iteration counter: $t = 1$

While ($t \leq G$) **do**

Make three groups from population in the same size and randomly select group particles.

Update Group 1:

For $i = 1$ to $Group1$ **do**

Update particle's position using eq(3).

End for

Update Group 2:

$r_1 = rand[0, 1]$.

$r_2 = rand[0, 1]$.

For $j = Group1 + 1$ to $Group2$ **do**

Update particle's position using eq (4).

End for

Update Group 3:

$r_3 = rand[0, 1]$.

For $k = Group2 + 1$ to $Group3$ **do**

Update particle's position using eq(5).

End for

Check Boundary Constraints

Get the current new location;

Evaluate the new positions and their fitness values.

If the new location is better than before, update it;

Rank the fitness values and find the current best individual X_{best} .

$t = t + 1$

End while

Return: X_{best}, f_{best}

2.2 Experimental Validation, Results and Discussion

To evaluate the convergence speed, accuracy, and stability of the TGES algorithm, the 26 benchmark functions ([11], [12], [13], [14] and [15]) (including unimodal, multimodal and fixed-dimensional functions) and CEC2017 test functions [16] were tested. Additionally, the performance of the TGES algorithm was compared with four well-regarded optimization algorithms (PSO [3], GWO [6], SCA [17], SSA [1] algorithm). It is important to highlight that PSO is one of the most traditional optimization techniques, while more recent algorithms like SSA, SCA, and GWO represent the cutting-edge approaches known for their superior performance. The details of the 26 benchmark test functions, including their expressions, dimensions, search ranges, and theoretical optimal fitness values f_{min} , are provided in Tables 1-3.

In order to ensure the fairness of the comparison and the reliability of the results, we set the experimental environments of all algorithms used for the comparison in the same way. Where, for the 26 benchmark test functions, the total number of particles in all test functions is set to $n = 60$, and the maximum number of iterations is set to $G = 1000$. For all functions in the experiment the dimensions

for F1-F18 are set to 60 as can see in table 1 and table 2. The experimental results being affected by randomness, each simulation experiment is repeated independently for 30 times to ensure that the experimental results are reliable. The parameter settings of each optimization algorithm are shown in Table 4. In addition, all the simulations are done in MATLAB R2021a platform and Windows 8.1 with CPU 2.80GHz and Ram 12GB.

This paper uses three statistical tools: mean, standard deviation and best value of the objective function. Their mathematical expressions are as follows:

$$Best = \min (f_1, f_2, \dots, f_N) \quad (7)$$

$$Mean = \frac{1}{N} \sum_{i=1}^N f_i \quad (8)$$

$$STD = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (f_i - Mean)^2} \quad (9)$$

where N is the number of optimization experiments, and f_i is the value of the test function for each independent run. $Mean$ is the average of the test function values in N tests, is one of the indicators for evaluating the convergence accuracy; STD is the standard deviation of the test function values in N tests, is an essential indicator for evaluating the stability of the algorithm; $Best$ is the best value of the test function values in N tests, is one of the indicators for evaluating the convergence accuracy.

Table 1: Description of unimodal benchmark functions.

Function	d	Range	f_{min}
$F_1(x) = \sum_{i=1}^d x_i^2$	60	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^d x_i + \prod_{i=1}^d x_i $	60	$[-10, 10]$	0
$F_3(x) = \sum_{i=1}^d \left(\sum_{j=1}^i x_j \right)^2$	60	$[-100, 100]$	0
$F_4(x) = \max_i \{ x_i , 1 \leq i \leq n \}$	60	$[-100, 100]$	0
$F_5(x) = \sum_{i=1}^d \left[(10^6)^{\frac{(i-1)}{(d-1)}} x_i^2 \right]$	60	$[-100, 100]$	0
$F_6(x) = \sum_{i=1}^d ix_i^4 + \text{random}[0,1]$	60	$[-1.28, 1.28]$	0
$F_7(x) = \sum_{i=1}^{d-1} (x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i^2+1)}$	60	$[-1, 4]$	0
$F_8(x) = \sum_{i=1}^d 100 (\text{random}[0,1])^i (\ln((x_i - 1)^2 + 1))^2$	60	$[-5, 5]$	0
$F_9(x) = \sum_{i=1}^{\frac{d}{4}} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$	60	$[-4, 5]$	0

Table 2: Description of multimodal benchmark functions.

Function	d	Range	f_{min}
$F_{10}(x) = \sum_{i=1}^d -x_i \sin(\sqrt{ x_i })$	60	[-500, 500]	-418.9829d
$F_{11}(x) = 10n + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$	60	[-5.12, 5.12]	0
$F_{12}(x) = -20e^{\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^d x_i^2}\right)} - e^{\left(\frac{1}{d}\sum_{i=1}^d \cos(2\pi x_i)\right)} + 20 + e$	60	[-32, 32]	0
$F_{13}(x) = \frac{1}{400} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	60	[-600, 600]	0
$F_{14}(x) = \sum_{i=1}^d x_i \sin(x_i) + 0.1x_i $	60	[-10, 10]	0
$F_{15}(x) = \left(1 + n - \sum_{i=1}^{d-1} x_i\right)^{d - \sum_{i=1}^{d-1} x_i}$	60	[0, 1]	2
$F_{16}(x) = \sum_{i=1}^d x_i^6 \left[2 + \sin\left(\frac{1}{x_i}\right)\right]$	60	[-1, 1]	0
$F_{17}(x) = \sum_{i=1}^d x_i^2 + \left(0.5 \sum_{i=1}^d ix_i\right)^2 + \left(0.5 \sum_{i=1}^d ix_i\right)^4$	60	[-5, 10]	0
$F_{18}(x) = \sum_{i=1}^{d-1} \left[\frac{\sin^2\left[\sqrt{x_{i+1}^2 + x_i^2}\right] - 0.5}{(0.001(x_{i+1}^2 + x_i^2) + 1)^2} + 0.5 \right]$	60	[-100, 100]	0

Table 3: Description of fixed dimension multimodal benchmark functions.

Function	d	Range	f_{min}
$F_{19}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5, 5]	0.0003075
$F_{20}(x) = x_1^2 - 100 \cos^2(x_1) - 100 \cos\left(\frac{x_1^2}{30}\right) + x_2^2 - 100 \cos^2(x_2) - 100 \cos\left(\frac{x_2^2}{30}\right)$	2	[-50, 50]	400
$f_{21}(x) = x_1^2 + x_2^2 + x_1 x_2 + \sin(x_1) + \cos(x_2) $	2	[-500, 500]	1
$f_{22}(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1 x_2$	2	[-10, 10]	0
$f_{23}(x) = 1 + \sin^2(x_1) + \sin^2(x_2) - 0.1e^{-(x_1^2 + x_2^2)}$	2	[-10, 10]	0.9
$f_{24}(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.3$	2	[-50, 50]	0
$f_{25}(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1 x_2 + x_2^2$	2	[-5, 5]	0
$f_{26}(x) = -\frac{0.001}{[(0.001)^2 + (x_1^2 + x_2^2 - 1)^2]} - \frac{[(0.001)^2 + (x_1^2 + x_2^2 - 0.5)^2]}{0.001} - \frac{[(0.001)^2 + (x_1^2 - x_2^2)^2]}{0.001}$	2	[-500, 500]	-2000

Table 4: Parameters settings for each algorithm.

algorithm	parameter	value
TGES	A, a	Linearly decrease from 1 to 0
	ω	1.5
	θ	$\pi/4$
PSO	inertia weight	Linearly decrease from 0.9 to 0.1
	c_1	2
	c_2	2
GWO	α_{min} and α_{max}	0 and 2

SCA	a	Linearly decrease from 2 to 0
SSA	PD	0.2
(Sparrow search algorithm)	ST	0.8
MFO	b	1
WOA	a	Linearly decrease from 2 to 0
HHO	E_0	$[-1, 1]$
	E_1	Linearly decrease from 2 to 0
GSA	α	20
	G_0	100
BAT	F_{min} and F_{max}	0 and 2
	α	0.5
	γ	0.5
ECO	H	0.5
	G_1, G_2	0.2, 0.1

2.3 TGES's performance on 26 benchmark test functions

For comparing the performance of algorithms on 26 benchmark test functions, performance metrics such as convergence speed and search capability can be obtained to measure and compare their strengths and weaknesses. By comparing the algorithms on the same benchmark test function, it can be determined which algorithm works better on a particular type of problem. Therefore, we have compared and tested the developed TGES algorithm with four excellent-performing optimization algorithms (PSO [3], GWO [6], SCA [17], and SSA [1] algorithms) on the 26 benchmark test functions. The table 5 shows the test results of TGES, PSO, GWO, SCA, and SSA algorithms on the 26 benchmark test functions.

Table 5: Comparison of optimization effect of each algorithm for F1-F26.

F	d		PSO	GWO	SCA	SSA	TGES
1	60	Best	4.01E-17	8.00E-54	3.68E-219	0.00E+00	0.00E+00
		Mean	4.87E-07	2.59E-52	1.95E-67	1.10E-121	0.00E+00
		STD	2.29E-06	4.23E-52	1.07E-66	6.01E-121	0.00E+00
2	60	Best	1.84E-02	6.28E-32	4.04E-112	0.00E+00	0.00E+00
		Mean	3.98E-01	2.71E-31	5.08E-58	1.51E-67	0.00E+00
		STD	3.20E-01	1.86E-31	2.59E-57	8.29E-67	0.00E+00
3	60	Best	2.16E+02	8.15E-04	1.29E+05	0.00E+00	0.00E+00
		Mean	6.45E+02	1.49E-01	2.54E+05	5.22E-76	0.00E+00
		STD	3.20E+02	4.32E-01	5.91E+04	2.86E-75	0.00E+00
4	60	Best	3.51E+00	2.85E-07	7.03E+01	0.00E+00	0.00E+00
		Mean	5.65E+00	4.65E-05	8.75E+01	2.63E-40	0.00E+00
		STD	9.66E-01	6.47E-05	4.02E+00	1.44E-39	0.00E+00
5	60	Best	6.97E-14	7.96E-51	5.93E-209	0.00E+00	0.00E+00
		Mean	3.73E-02	2.12E-49	1.17E-33	2.10E-116	0.00E+00
		STD	1.90E-01	2.51E-49	6.42E-33	1.15E-115	0.00E+00
6	60	Best	2.49E-02	8.28E-04	4.81E-06	3.29E-05	1.75E-06
		Mean	4.37E-02	1.88E-03	1.53E+00	9.31E-04	2.66E-05
		STD	1.23E-02	5.67E-04	6.88E+00	8.31E-04	2.28E-05
7	60	Best	1.12E-18	3.77E-54	3.50E-224	0.00E+00	0.00E+00
		Mean	4.38E-07	8.80E-48	1.31E-68	2.97E-115	0.00E+00
		STD	2.39E-06	4.48E-47	7.19E-68	1.63E-114	0.00E+00
8	60	Best	8.89E-02	6.08E-01	2.43E-04	9.62E-01	1.19E-02
		Mean	3.68E+00	4.49E+00	1.39E-01	4.99E+00	6.85E-02
		STD	3.82E+00	3.05E+00	2.05E-01	2.44E+00	5.66E-02
9	60	Best	8.00E-03	3.38E-05	9.66E-200	0.00E+00	0.00E+00

F	d		PSO	GWO	SCA	SSA	TGES
		Mean	4.93E-02	2.40E-04	5.22E-36	4.34E-112	0.00E+00
		STD	4.40E-02	2.56E-04	2.86E-35	2.38E-111	0.00E+00
10	60	Best	-1.53E+04	-1.45E+04	-2.51E+04	-2.02E+04	-2.51E+04
		Mean	-1.29E+04	-1.12E+04	-1.95E+04	-1.81E+04	-2.26E+04
		STD	1.47E+03	3.17E+03	3.13E+03	9.55E+02	3.09E+03
11	60	Best	5.57E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
		Mean	9.98E+01	1.67E+01	0.00E+00	0.00E+00	0.00E+00
		STD	2.24E+01	8.63E+00	0.00E+00	0.00E+00	0.00E+00
12	60	Best	1.34E+00	4.44E-15	8.88E-16	8.88E-16	8.88E-16
		Mean	2.23E+00	4.44E-15	8.88E-16	8.88E-16	8.88E-16
		STD	5.41E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
13	60	Best	9.14E-14	0.00E+00	0.00E+00	0.00E+00	0.00E+00
		Mean	4.51E-02	1.69E-03	0.00E+00	0.00E+00	0.00E+00
		STD	7.13E-02	4.45E-03	0.00E+00	0.00E+00	0.00E+00
14	60	Best	8.90E-03	2.93E-29	5.31E-101	0.00E+00	0.00E+00
		Mean	7.74E-02	1.14E-03	5.14E-53	4.46E-66	0.00E+00
		STD	9.59E-02	1.22E-03	2.82E-52	2.44E-65	0.00E+00
15	60	Best	2.00E+00	2.30E+00	2.00E+00	2.00E+00	2.00E+00
		Mean	2.00E+00	4.57E+02	2.00E+00	7.54E+07	2.00E+00
		STD	2.54E-12	1.76E+03	0.00E+00	2.52E+08	0.00E+00
16	60	Best	1.07E-08	1.09E-107	0.00E+00	0.00E+00	0.00E+00
		Mean	2.26E-07	1.46E-99	2.50E-01	7.10E-261	0.00E+00
		STD	2.30E-07	6.97E-99	6.43E-01	0.00E+00	0.00E+00
17	60	Best	8.85E+00	1.55E-07	7.69E-76	0.00E+00	0.00E+00
		Mean	1.72E+01	2.04E-05	2.44E+02	1.28E-49	0.00E+00
		STD	7.13E+00	2.93E-05	1.85E+02	6.99E-49	0.00E+00
18	60	Best	1.49E+01	1.50E+01	0.00E+00	0.00E+00	0.00E+00
		Mean	1.89E+01	2.22E+01	0.00E+00	0.00E+00	0.00E+00
		STD	2.07E+00	2.99E+00	0.00E+00	0.00E+00	0.00E+00
19	4	Best	3.07E-04	3.07E-04	3.09E-04	3.07E-04	3.07E-04
		Mean	1.88E-03	3.01E-03	9.71E-04	5.03E-04	3.52E-04
		STD	5.05E-03	6.92E-03	1.46E-03	3.33E-04	1.69E-04
20	2	Best	-4.00E+02	-4.00E+02	-4.00E+02	-4.00E+02	-4.00E+02
		Mean	-4.00E+02	-4.00E+02	-4.00E+02	-4.00E+02	-4.00E+02
		STD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
21	2	Best	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
		Mean	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
		STD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
22	2	Best	0.00E+00	0.00E+00	3.19E-273	0.00E+00	0.00E+00
		Mean	0.00E+00	0.00E+00	1.43E-200	1.59E-179	0.00E+00
		STD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
23	2	Best	9.00E-01	9.00E-01	9.00E-01	9.00E-01	9.00E-01
		Mean	9.13E-01	9.00E-01	9.00E-01	9.00E-01	9.00E-01
		STD	3.46E-02	4.52E-16	2.97E-15	4.52E-16	4.52E-16
24	2	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
		Mean	0.00E+00	0.00E+00	2.67E-08	0.00E+00	0.00E+00
		STD	0.00E+00	0.00E+00	1.07E-07	0.00E+00	0.00E+00

F	d		PSO	GWO	SCA	SSA	TGES
25	2	Best	0.00E+00	0.00E+00	1.06E-250	0.00E+00	0.00E+00
		Mean	0.00E+00	0.00E+00	2.08E-15	1.33E-251	0.00E+00
		STD	0.00E+00	0.00E+00	1.14E-14	0.00E+00	0.00E+00
26	2	Best	-2.00E+03	-1.00E+03	-2.00E+03	-2.00E+03	-2.00E+03
		Mean	-1.10E+03	-1.00E+03	-1.99E+03	-1.42E+03	-1.99E+03
		STD	3.05E+02	2.52E-03	1.82E+01	4.06E+02	1.80E+01

After testing all algorithms on selected 26 test functions, as can be seen in table 5, it shows that the TGES algorithm is slightly better than the other four algorithms from the average values obtained for all test functions. And the TGES algorithm got optimal value for (F1-F5, F7, F11, F13-F18, F20-F22, F24 and F25). In order to better describe the optimization process and convergence speed of each algorithm, the convergence graph of each algorithm on each function is given, as shown in Figure 1-3. As shown in Figure 1-3, TGES has great advantages in the optimization speed and convergence accuracy of each function. It converges quickly on the unimodal function and on the multimodal function and it has good convergence on the fixed-dimensional function. It is evident that the TGES algorithm achieved efficient solution stability on the 26 test problems.

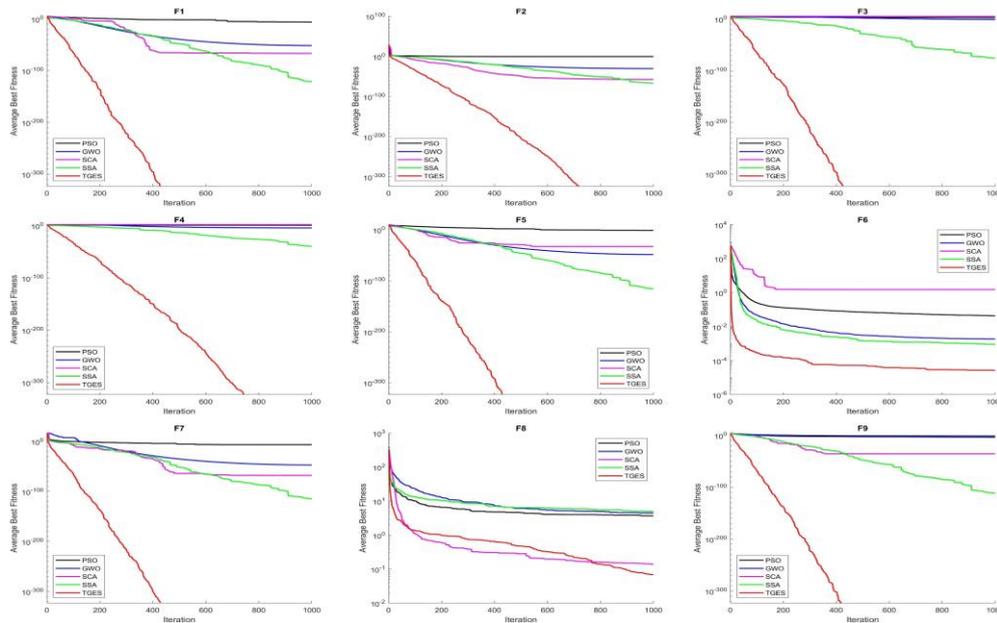


Fig. 1: The convergence curve of the five algorithms on the unimodal test functions F1-F9.

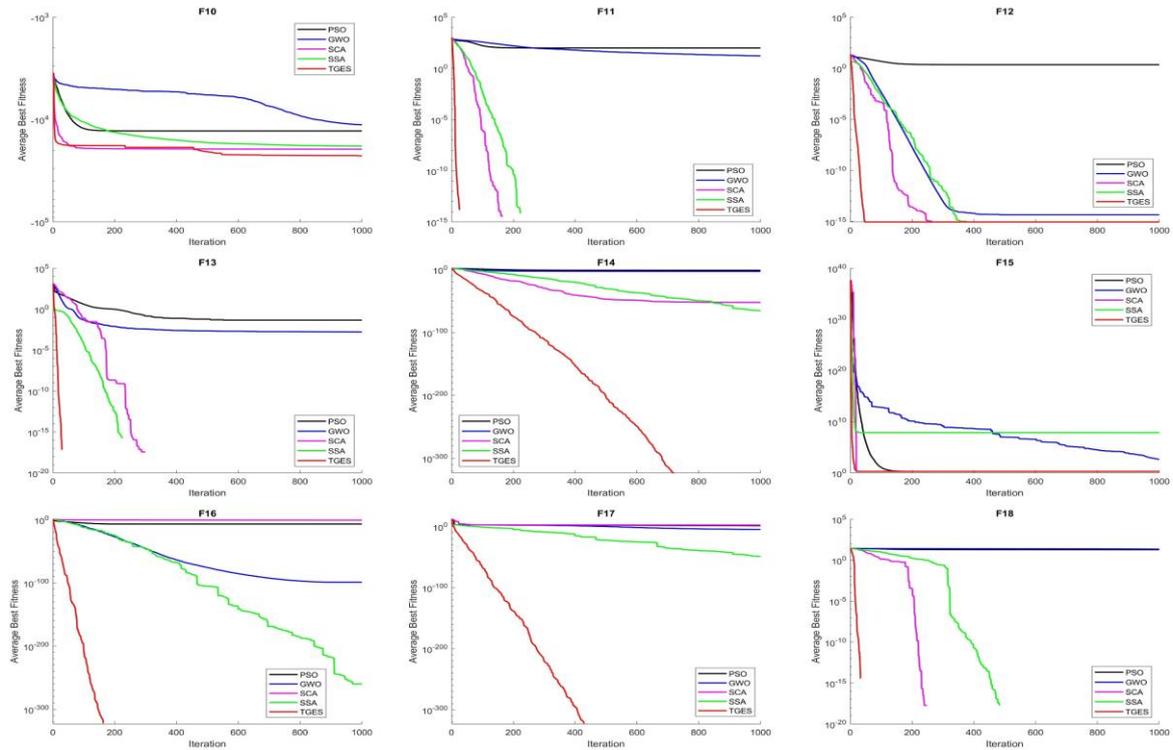


Fig. 2: The convergence curve of the five algorithms on the multimodal test functions F10-F18.

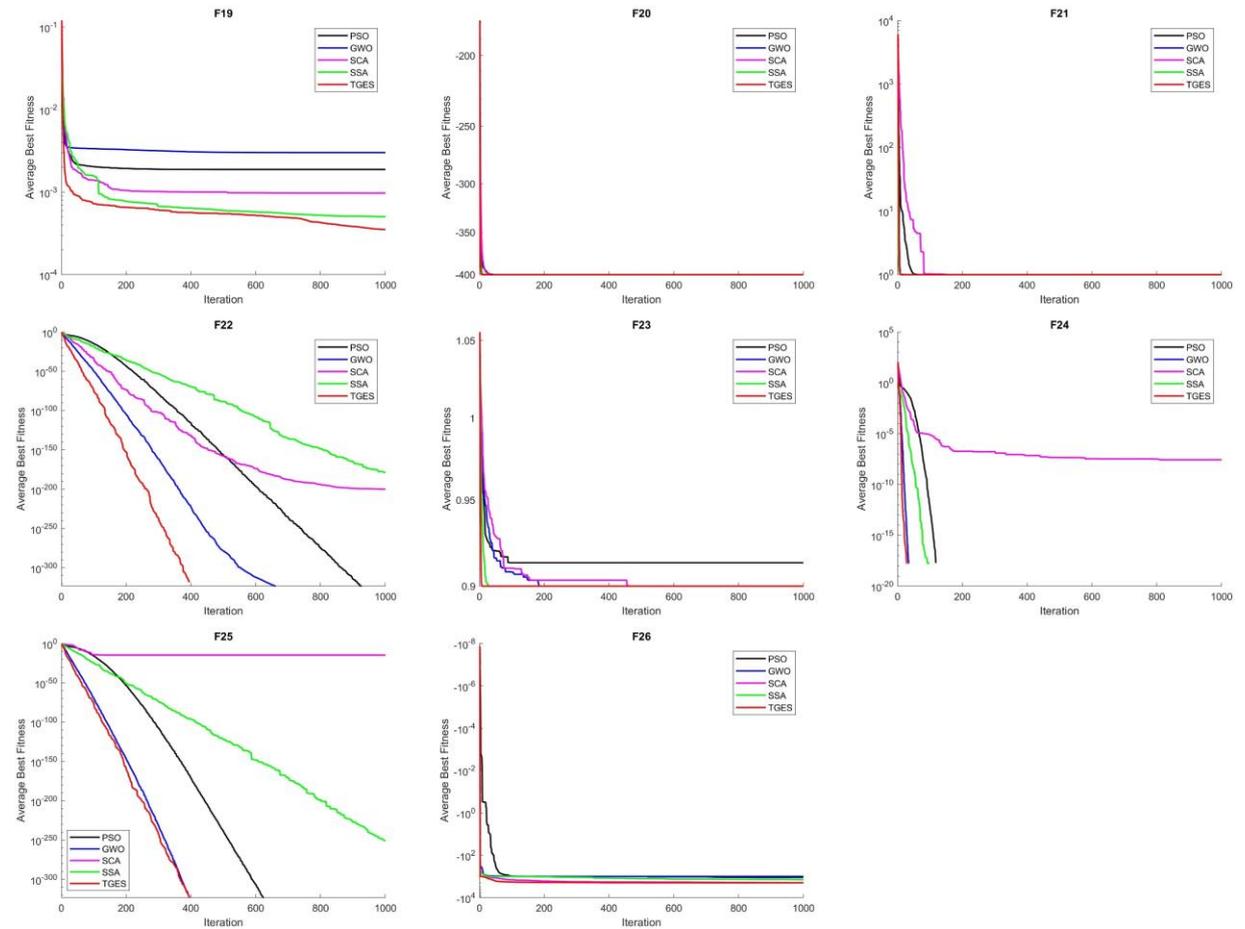


Fig. 3: The convergence curve of the five algorithms on the fixed-dimension test functions F19-F26.

2.4 TGES's performance on IEEE CEC-2017 test functions

To verify the processing capability of the TGES algorithm for complex feature functions and its applicability to different complex optimization problems, this study used the IEEE CEC-2017 test function [16] to test its performance. The CEC-2017 test function with their dimensions and global minimum of each function was introduced as shown in Table 6. In this paper, the TGES algorithm was compared with the (PSO [3], GWO [6], SCA [17], and SSA [1] algorithms) for the performance of the search. The experimental parameters were taken as population size $n = 60$, dimension $d = 30$, and the maximum number of iterations $G = 1000$, and each function was run 30 times independently to obtain the best fitness, mean and standard deviation. The comparison of the results of each algorithm run is shown in Table 7.

Table 6: IEEE CEC-2017 benchmark test functions.

No.	Function	d	f_{min}
CEC-1	Shifted and Rotated Bent Cigar Function	30	100
CEC-2	Shifted and Rotated Sum of Different Power Function	30	200
CEC-3	Shifted and Rotated Zakharov Function	30	300
CEC-4	Shifted and Rotated Rosenbrock's Function	30	400
CEC-5	Shifted and Rotated Rastrigin's Function	30	500
CEC-6	Shifted and Rotated Expanded Scaffer's Function	30	600
CEC-7	Shifted and Rotated Lunacek Bi_Rastrigin Function	30	700
CEC-8	Shifted and Rotated Non-Continuous Rastrigin's Function	30	800
CEC-9	Shifted and Rotated Levy Function	30	900
CEC-10	Shifted and Rotated Schwefel's Function	30	1000
CEC-11	Hybrid Function 1 (N = 3)	30	1100
CEC-12	Hybrid Function 2 (N = 3)	30	1200
CEC-13	Hybrid Function 3 (N = 3)	30	1300
CEC-14	Hybrid Function 4 (N = 4)	30	1400
CEC-15	Hybrid Function 5 (N = 4)	30	1500
CEC-16	Hybrid Function 6 (N = 4)	30	1600
CEC-17	Hybrid Function 6 (N = 5)	30	1700
CEC-18	Hybrid Function 6 (N = 5)	30	1800
CEC-19	Hybrid Function 6 (N = 5)	30	1900
CEC-20	Hybrid Function 6 (N = 6)	30	2000
CEC-21	Composition Function 1 (N = 3)	30	2100
CEC-22	Composition Function 2 (N = 3)	30	2200
CEC-23	Composition Function 3 (N = 4)	30	2300
CEC-24	Composition Function 4 (N = 4)	30	2400
CEC-25	Composition Function 5 (N = 5)	30	2500
CEC-26	Composition Function 6 (N = 5)	30	2600
CEC-27	Composition Function 7 (N = 6)	30	2700
CEC-28	Composition Function 8 (N = 6)	30	2800
CEC-29	Composition Function 9 (N = 3)	30	2900
CEC-30	Composition Function 10 (N = 3)	30	3000

Table 7: Comparison of optimization effect of each algorithm on CEC-2017 benchmark test functions.

F		PSO	GWO	SCA	SSA	TGES
CEC-1	Best	4.11E+09	4.60E+07	1.16E+10	5.34E+03	7.10E+08
	Mean	1.71E+10	5.82E+08	2.21E+10	1.12E+05	4.12E+09
	STD	8.82E+09	8.70E+08	6.26E+09	1.79E+05	2.94E+09

CEC-2	Best	2.78E+25	4.87E+15	3.10E+31	5.45E+16	2.98E+19
	Mean	8.38E+37	8.39E+27	3.74E+40	3.02E+22	3.24E+29
	STD	4.48E+38	4.59E+28	1.98E+41	9.68E+22	1.16E+30
CEC-3	Best	6.12E+04	1.99E+03	7.54E+04	3.84E+04	1.55E+04
	Mean	1.33E+05	8.33E+03	1.16E+05	5.77E+04	2.66E+04
	STD	4.90E+04	4.43E+03	2.41E+04	8.31E+03	6.38E+03
CEC-4	Best	7.20E+02	5.05E+02	2.36E+03	4.73E+02	5.57E+02
	Mean	3.19E+03	5.41E+02	4.87E+03	5.05E+02	6.86E+02
	STD	2.01E+03	3.07E+01	1.93E+03	2.44E+01	9.61E+01
CEC-5	Best	5.68E+02	5.46E+02	7.22E+02	5.61E+02	6.17E+02
	Mean	6.77E+02	6.09E+02	8.06E+02	6.31E+02	6.83E+02
	STD	4.85E+01	6.61E+01	3.83E+01	3.39E+01	3.27E+01
CEC-6	Best	6.08E+02	6.01E+02	6.39E+02	6.02E+02	6.18E+02
	Mean	6.27E+02	6.04E+02	6.57E+02	6.16E+02	6.37E+02
	STD	1.11E+01	2.06E+00	1.00E+01	1.33E+01	1.13E+01
CEC-7	Best	8.22E+02	7.84E+02	1.08E+03	8.51E+02	9.02E+02
	Mean	1.04E+03	8.61E+02	1.24E+03	9.38E+02	9.88E+02
	STD	1.26E+02	5.48E+01	8.50E+01	6.62E+01	4.86E+01
CEC-8	Best	8.71E+02	8.28E+02	9.97E+02	8.78E+02	8.98E+02
	Mean	9.80E+02	8.92E+02	1.05E+03	9.30E+02	9.59E+02
	STD	4.99E+01	5.10E+01	2.87E+01	3.54E+01	3.01E+01
CEC-9	Best	1.65E+03	9.93E+02	5.04E+03	1.54E+03	2.65E+03
	Mean	7.28E+03	1.35E+03	6.90E+03	5.62E+03	4.14E+03
	STD	3.16E+03	3.63E+02	1.19E+03	1.92E+03	1.22E+03
CEC-10	Best	3.93E+03	3.37E+03	6.67E+03	4.21E+03	4.15E+03
	Mean	5.44E+03	6.31E+03	7.87E+03	6.30E+03	5.85E+03
	STD	8.49E+02	2.29E+03	5.41E+02	1.40E+03	7.83E+02
CEC-11	Best	1.27E+03	1.23E+03	4.22E+03	1.23E+03	1.31E+03
	Mean	1.63E+03	1.29E+03	9.83E+03	1.36E+03	1.44E+03
	STD	2.85E+02	4.49E+01	3.57E+03	7.13E+01	1.25E+02
CEC-12	Best	2.24E+07	3.20E+06	8.83E+08	2.29E+05	8.60E+06
	Mean	1.31E+09	2.94E+07	2.36E+09	1.68E+06	9.59E+07
	STD	1.24E+09	2.65E+07	1.12E+09	1.27E+06	9.66E+07
CEC-13	Best	2.81E+04	3.74E+04	2.55E+06	8.54E+03	1.38E+05
	Mean	7.51E+08	1.60E+05	4.11E+08	1.08E+05	3.07E+07
	STD	1.19E+09	2.22E+05	4.10E+08	7.43E+04	8.58E+07
CEC-14	Best	1.25E+04	4.64E+03	7.25E+04	3.95E+03	2.80E+03
	Mean	2.94E+05	5.83E+04	1.15E+06	6.63E+04	1.29E+05
	STD	4.16E+05	4.51E+04	1.20E+06	6.76E+04	1.32E+05
CEC-15	Best	1.31E+04	8.30E+03	1.49E+06	2.61E+03	3.66E+04
	Mean	1.15E+05	6.96E+04	5.10E+07	4.30E+04	3.54E+05
	STD	1.13E+05	5.29E+04	1.28E+08	5.16E+04	4.71E+05
CEC-16	Best	2.36E+03	1.92E+03	3.14E+03	2.11E+03	2.05E+03
	Mean	3.22E+03	2.60E+03	3.88E+03	2.69E+03	2.72E+03
	STD	4.61E+02	4.85E+02	3.50E+02	3.17E+02	3.05E+02

CEC-17	Best	2.09E+03	1.79E+03	2.15E+03	1.91E+03	1.80E+03
	Mean	2.57E+03	1.94E+03	2.54E+03	2.32E+03	2.04E+03
	STD	2.56E+02	1.29E+02	2.91E+02	2.72E+02	1.59E+02
CEC-18	Best	1.25E+05	1.36E+05	1.29E+05	4.67E+04	7.42E+04
	Mean	2.40E+06	7.12E+05	9.01E+06	9.86E+05	7.37E+05
	STD	2.19E+06	7.13E+05	1.00E+07	1.16E+06	8.05E+05
CEC-19	Best	2.68E+04	3.68E+03	4.09E+06	2.36E+03	1.23E+04
	Mean	4.24E+07	5.63E+05	6.23E+07	3.09E+04	1.44E+06
	STD	6.08E+07	6.35E+05	7.15E+07	4.36E+04	1.73E+06
CEC-20	Best	2.23E+03	2.17E+03	2.41E+03	2.15E+03	2.26E+03
	Mean	2.56E+03	2.36E+03	2.69E+03	2.51E+03	2.42E+03
	STD	1.85E+02	1.26E+02	1.07E+02	1.76E+02	1.07E+02
CEC-21	Best	2.44E+03	2.34E+03	2.52E+03	2.38E+03	2.39E+03
	Mean	2.50E+03	2.41E+03	2.58E+03	2.43E+03	2.43E+03
	STD	3.49E+01	6.54E+01	3.33E+01	4.85E+01	2.62E+01
CEC-22	Best	3.01E+03	2.35E+03	3.80E+03	2.30E+03	2.44E+03
	Mean	6.43E+03	4.59E+03	7.37E+03	3.34E+03	3.25E+03
	STD	1.77E+03	2.53E+03	2.18E+03	1.96E+03	1.65E+03
CEC-23	Best	2.85E+03	2.70E+03	2.94E+03	2.70E+03	2.75E+03
	Mean	3.00E+03	2.74E+03	3.14E+03	2.77E+03	2.82E+03
	STD	9.37E+01	4.45E+01	9.94E+01	3.50E+01	3.61E+01
CEC-24	Best	3.06E+03	2.87E+03	3.07E+03	2.87E+03	2.92E+03
	Mean	3.19E+03	2.93E+03	3.33E+03	2.96E+03	2.98E+03
	STD	8.72E+01	6.60E+01	1.42E+02	4.94E+01	4.26E+01
CEC-25	Best	2.92E+03	2.89E+03	3.22E+03	2.88E+03	2.93E+03
	Mean	3.53E+03	2.94E+03	3.68E+03	2.90E+03	3.01E+03
	STD	5.69E+02	1.86E+01	3.05E+02	1.53E+01	4.25E+01
CEC-26	Best	5.59E+03	4.04E+03	6.44E+03	4.00E+03	3.27E+03
	Mean	7.33E+03	4.64E+03	8.04E+03	5.18E+03	4.96E+03
	STD	1.01E+03	5.43E+02	1.10E+03	5.10E+02	1.11E+03
CEC-27	Best	3.26E+03	3.21E+03	3.37E+03	3.21E+03	3.21E+03
	Mean	3.36E+03	3.23E+03	3.58E+03	3.25E+03	3.27E+03
	STD	9.02E+01	1.69E+01	1.49E+02	2.30E+01	3.39E+01
CEC-28	Best	3.51E+03	3.24E+03	3.93E+03	3.21E+03	3.34E+03
	Mean	5.21E+03	3.33E+03	5.16E+03	3.26E+03	3.45E+03
	STD	1.24E+03	6.69E+01	5.99E+02	4.91E+01	8.15E+01
CEC-29	Best	3.68E+03	3.45E+03	3.94E+03	3.54E+03	3.70E+03
	Mean	4.35E+03	3.69E+03	5.08E+03	4.06E+03	4.02E+03
	STD	4.27E+02	1.72E+02	6.46E+02	3.15E+02	2.37E+02
CEC-30	Best	2.84E+04	5.73E+05	8.90E+06	1.07E+04	1.08E+06
	Mean	8.45E+06	5.08E+06	7.05E+07	2.72E+05	1.18E+07
	STD	7.33E+06	3.77E+06	7.69E+07	3.82E+05	8.99E+06

From the results in Table 7, a comparison of the TGES algorithm and four other algorithms on CEC-2017 benchmark test functions, based on mean values over 30 runs, shows that the TGES algorithm recorded:

1. Rank 1 for CEC-22.
2. Rank 2 for (CEC-3, CEC-9, CEC-10, CEC-17, CEC-18, CEC-20, CEC-26 and CEC-29).
3. Rank 3 for (CEC-1, CEC-2, CEC-4, CEC-7, CEC-8, CEC-11, CEC-12, CEC-13, CEC-14, CEC-16, CEC-19, CEC-21, CEC-23, CEC-24, CEC-25, CEC-27 and CEC-28).
4. Rank 4 for (CEC-5, CEC-6, CEC-15, and CEC-30).

Based on the results for the CEC-2017 test functions, it can be concluded that the TGES algorithm successfully solved these problems with strong performance.

3.0 ENGINEERING APPLICATION PROBLEMS

In this section, the six practical engineering problems are chosen to illustrate the competitiveness of the TGES algorithm in solving constrained optimization problems with mix variables. For the processing of the inequality constraints in the problem we use penalty functions, which embeds constraints into the objective function. Before the experiment, we set the parameters of the TGES algorithm: total number of particles $n = 130$, and maximum number of iterations $G = 800$.

3.1 Optimization of design problem for industrial refrigeration systems

At present, energy saving and emission reduction work has become the focus of various fields. Industrial refrigeration system accounts for a large proportion of energy consumption, so it is necessary to optimize and control the industrial refrigeration system. Optimal design of industrial refrigeration system is an extremely complex engineering design problem, which has fourteen design variables and fifteen constraints. Its mathematical model is shown as follows:

Consider $x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}]$
 minimize $f(x) = 63098.88s_2x_4x_{12} + 5441.5x_2^2x_{12} + 115055.5x_2^{1.664}s_6 + 6172.27x_2^2x_6 + 63098.88s_1x_3x_{11}$
 $+ 5441.5x_1^2x_{11} + 115055.5x_1^{1.644}x_5 + 6172.27x_1^2x_5 + 140.53x_1x_{11} + 281.29x_3x_{11}$
 $+ 70.26x_1^2 + 281.29x_1x_3 + 281.29x_3^2 + 20470.2x_7^{2.893}x_{11}^{0.316}x_1^2$
 $+ 14437x_8^{1.8812}x_{12}^{0.3424}x_{10}x_{14}^{-1}x_1^2x_7x_9^{-1}$

Subject to $g_1(x) = 1.524x_7^{-1} - 1 \leq 0,$
 $g_2(x) = 1.524x_8^{-1} - 1 \leq 0,$
 $g_3(x) = 0.07789x_1 - 2x_7^{-1}x_9 - 1 \leq 0,$
 $g_4(x) = 7.05305s_9^{-1}x_1^2x_{10}x_8^{-1}x_2^{-1}x_{14} - 1 \leq 0,$
 $g_5(x) = \frac{0.0833}{x_{13}}x_{14} - 1 \leq 0,$
 $g_6(x) = 0.04771x_{10}x_8^{1.8812}x_{12}^{0.3424} - 1 \leq 0,$
 $g_7(x) = 0.0488x_9x_7^{1.893}x_{11}^{0.316} - 1 \leq 0,$
 $g_8(x) = \frac{0.0099x_1}{x_3} - 1 \leq 0,$
 $g_9(x) = \frac{0.0193x_2}{x_4} - 1 \leq 0,$
 $g_{10}(x) = \frac{0.0298x_1}{x_5} - 1 \leq 0,$
 $g_{11}(x) = \frac{47.136x_2x_{12}}{x_{10}} - 1.333x_8x_{13}^{2.1195} + \frac{62.08x_{13}^{2.1195}x_8^2}{x_{12}x_{10}} - 1 \leq 0,$
 $g_{12}(x) = \frac{0.056x_2}{x_6} - 1 \leq 0,$
 $g_{13}(x) = \frac{2}{x_9} - 1 \leq 0,$

Variable range: $0.001 < x_i < 5, i = 1, \dots, 14$

Fourteen key variables of optimal design of industrial refrigeration system are optimized by TGES algorithm, and the results are compared with (HHO [18], WOA [7], SSA [1] and ECO [19] algorithms).

Table 8 shows the lowest cost of each algorithm and the values of related variables for optimal design of industrial refrigeration system and it shows that TGES algorithm ranks first among other algorithms.

Table 8: Comparison of result on optimal design of industrial refrigeration system.

Algorithm	TGES	HHO	WOA	SSA	ECO
x_1	0.0010	0.0010	0.0010	0.0010	0.0010
x_2	0.0011	0.0010	0.0011	0.0010	0.0010
x_3	0.0010	0.0010	0.0011	0.0010	0.0010
x_4	0.0122	1.3382	0.0012	0.0010	0.0010
x_5	0.0011	1.7791	0.0011	0.0010	0.0010
x_6	0.0022	1.5272	0.0011	0.0010	0.0010
x_7	1.5276	1.5997	1.5642	1.5240	1.5244
x_8	1.5333	2.2739	1.5240	1.5240	1.5240
x_9	5.0000	2.2525	4.9997	5.0000	5.0000
x_{10}	2.0275	2.0000	2.0267	2.0000	3.4759
x_{11}	0.0012	1.7746	0.0019	0.0010	0.0010
x_{12}	0.0011	0.0010	0.0011	0.0010	0.0010
x_{13}	0.0075	0.0040	0.0076	0.0010	0.0089
x_{14}	0.0899	0.0289	0.0906	0.0120	0.1067
Fitness	0.03629	5.5671	0.036694	0.16365	0.0413

3.2 Pressure vessel design problem

In the process of pressure vessel design, the selection of parameters will directly affect the performance and reliability of the vessel. Adjusting parameters x_1 , x_2 , x_3 and x_4 will impact the structural strength, durability and ability of the container to withstand pressure. However, several constraints, such as those represented by g_1 , g_2 , g_3 and g_4 , must be taken into account when optimizing these parameters to ensure the safe operation of the vessel under various operating conditions and the constraint variables are shown in Figure 4. This requires introducing engineering mathematical models and optimization algorithms into the design process to balance the interaction of parameters and meet constraints. In summary, the design of pressure vessels seeks to minimize manufacturing costs by optimizing critical parameters under a range of constraints [20]. Its engineering mathematical model is as follows:

Consider $x = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L]$

minimize $f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$

Subject to $g_1(x) = -x_1 + 0.0193x_3 \leq 0$,

$g_2(x) = -x_2 + 0.00954x_3 \leq 0$,

$g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0$,

$g_4(x) = x_4 - 240 \leq 0$,

Variable range: $0 \leq x_1 \leq 99, 0 \leq x_2 \leq 99, 10 \leq x_3 \leq 200, 10 \leq x_4 \leq 200$

We compare the optimization results of the TGES algorithm with the solution results of (HHO [18], GSA [21], SCA [17], BAT [22], WOA [7], PSO [3], SSA [1] and ECO [19] algorithms). As can be seen from Table 9, TGES's optimization ability ranks first among these algorithms in dealing with pressure vessel design problems.

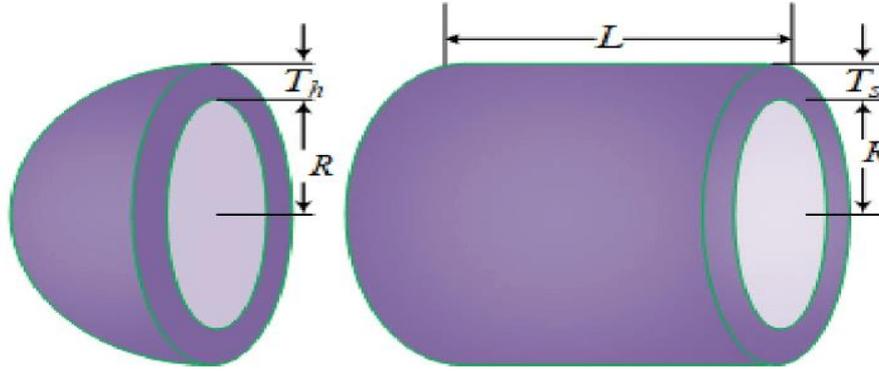


Fig. 4: Pressure vessel design problem.

Table 9: Comparison of result on pressure vessel design problem.

Algorithm	x_1	x_2	x_3	x_4	Fitness
TGES	0.779270582	0.388160006	40.3580885	199.4756066	5898.535847
HHO	0.92983957	0.456605095	47.86216883	116.2658059	6219.637266
GSA	2.038983402	1.007870552	105.6468084	115.097347	45662.57436
SCA	1.314512396	0.654039793	65.2381419	10	7774.500038
BAT	1.058928368	1.175846414	54.8667186	165.4201672	14083.70575
WOA	1.116965728	0.524071378	53.77330954	70.96863748	6958.906957
PSO	0.822992188	0.406805473	42.64208226	170.014545	5966.449084
SSA	0.802335113	0.396594662	41.57176752	183.274126	5927.937369
ECO	1.084417134	0.536027953	56.18741625	55.75378024	6641.853623

3.3 Speed reducer design problem

Speed reducer problem is an engineering problem with complex constraints, and its optimization purpose is to minimize the weight of speed reducer itself. The constraint variables are shown in Figure 5. The mathematical model of speed reducer is as follows:

Consider $x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [b, m, z, l_1, l_2, d_1, d_2]$

$$\text{minimize } f(x) = 0.7894x_2^2x_1(14.9334x_3 - 43.0934 + 3.3333x_2^2) + 0.7854(x_5x_7^2 + x_4x_6^2) - 1.508x_1(x_7^2 + x_6^2) + 7.477(x_7^3 + x_6^3)$$

$$\text{Subject to } g_1(x) = -x_1x_2^2x_3 + 27 \leq 0,$$

$$g_2(x) = -x_1x_2^2x_3^2 + 397.5 \leq 0,$$

$$g_3(x) = -x_2x_6^4x_3x_4^{-3} + 1.93 \leq 0,$$

$$g_4(x) = -x_2x_7^4x_3x_5^{-3} + 1.93 \leq 0,$$

$$g_5(x) = 10x_6^{-3} \sqrt{16.91 \times 10^6 + (745x_4x_2^{-1}x_3^{-1})^2} - 1100 \leq 0,$$

$$g_6(x) = 10x_7^{-3} \sqrt{157.5 \times 10^6 + (745x_5x_2^{-1}x_3^{-1})^2} - 850 \leq 0,$$

$$g_7(x) = x_2x_3 - 40 \leq 0,$$

$$g_8(x) = -x_1x_2^{-1} + 5 \leq 0,$$

$$g_9(x) = x_1x_2^{-1} - 12 \leq 0,$$

$$g_{10}(x) = 1.5x_6 - x_4 + 1.9 \leq 0,$$

$$g_{11}(x) = 1.1x_7 - x_5 + 1.9 \leq 0,$$

Variable range: $2.6 \leq x_1 \leq 3.6$, $0.7 \leq x_2 \leq 0.8$, $17 \leq x_3 \leq 28$, $7.3 \leq x_4$, $x_5 \leq 8.3$, $2.9 \leq x_6 \leq 3.9$, $5 \leq x_7 \leq 5.5$.

On the basis of improved algorithm TGES, the speed reducer problem is optimized and the values of relevant parameters are obtained. The optimization results are compared with (HHO [18], GSA [21], SCA [17], BAT [22], WOA [7], GWO [6] and ECO [19] algorithms). The details are shown in Table 10. To more clearly reflect the optimal cost of each algorithm. Compared with other algorithms, the TGES algorithm in this paper has higher accuracy in dealing with speed reducer engineering problem. In other words, The TGES algorithm find the best values for seven design variables to minimize the weight of speed reducer.

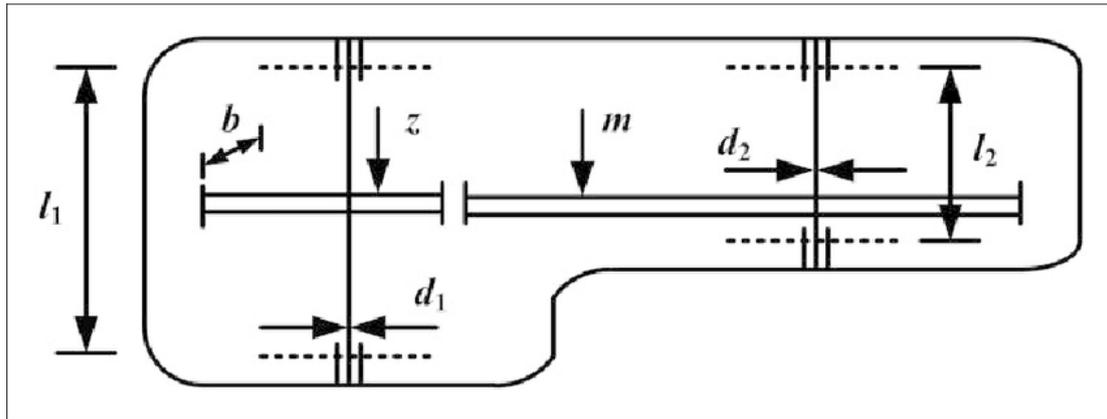


Fig. 5: Speed reducer design problem.

Table 10: Comparison of result on speed reducer design problem

Algorit hm	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Fitness
TGES	3.500433 064	0.7	17	7.314250 785	7.879188 666	3.362163 475	5.288960 906	3010.814 581
HHO	3.5	0.7	17	7.588784 877	7.757030 636	3.792503 223	5.305784 396	3146.913 4
GSA	3.567998 169	0.700027 733	22.45795 558	7.730575 549	7.869031 652	3.614968 175	5.354945 008	4257.152 353
SCA	3.547046 049	0.715504 325	18.17173 132	7.638240 185	7.861843 422	3.779277 497	5.419098 13	42596672 .91
BAT	3.439192 349	0.675139 265	18.43359 909	7.374066 955	7.776105 543	3.565781 168	5.286561 748	3166.096 837
WOA	3.506027 693	0.7	17	7.3	7.715444 053	3.445358 78	5.286654 843	3029.709 162
GWO	3.502612 109	0.7	17	7.835169 451	8.006692 71	3.353613 641	5.289095 902	3016.974 929
ECO	3.500000 668	0.700000 032	17.00018 481	8.286573 566	7.715322 302	3.352503 898	5.286655 581	3011.721 434

3.4 Three-bar truss design problem

The problem of designing a three-pole truss is not just a simple parameter adjustment process but a complex task that takes into account knowledge in the fields of engineering, mathematical optimization and material mechanics. It is necessary to find the optimal design scheme through appropriate mathematical modeling and optimization methods under the premise of satisfying multiple constraints. This problem involves adjusting two parameters (x_1, x_2) to achieve the minimum structural weight while satisfying three constraints (g_1, g_2, g_3) [23]. The constraint variables are shown in Figure 6 and its mathematical expression is as follows:

Consider $x = [x_1, x_2] = [A_1, A_2]$

minimize $f(x) = (2\sqrt{2}x_1 + x_2) \times l$

Subject to $g_1(x) = \frac{(\sqrt{2}x_1 + x_2)}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0,$

$g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0,$

$g_3(x) = \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \leq 0$

Variable range: $0 \leq x_1, x_2 \leq 1$,
 $l = 100\text{cm}$, $P = 2\text{ km/cm}^2$, $\sigma = 2\text{ km/cm}_2$.

By comparing the solution results of the TGES algorithm with the optimization results of the (HHO [18], GSA [21], SCA [17], BAT [22], WOA [7] and MFO [24] algorithms), it can be found that the TGES algorithm has better performance in the three-bar truss design problem. as shown in Table 11, the TGES algorithm proposed in this paper shows a faster solution speed and Higher solution accuracy compared with the other algorithms. Therefore, the algorithm can effectively handle this class of constraint problems.

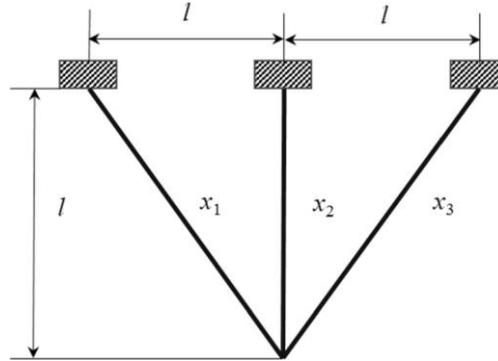


Figure 6: Three-bar truss design problem.

Table 11: Comparison of result on three-bar truss design problem.

Algorithm	x_1	x_2	Fitness
TGES	0.789814957	0.405040638	263.8974685
HHO	0.795687644	0.388766712	263.9311227
GSA	0.743798206	0.552702961	265.6481981
SCA	0.789929944	0.404733715	263.8992994
BAT	0.823517183	0.317661793	264.692013
WOA	0.79569787	0.388739539	263.9312977
MFO	0.79281031	0.396676129	263.9082316

3.5 Welded beam design problem

The main goal of this problem is to find the optimal solution to the welded beam design manufacturing cost. It is worth highlighting that the parameters involved in the welded beam design problem include four key factors, namely x_1, x_2, x_3 and x_4 , accompanied by seven constraints, namely $g_1, g_2, g_3, g_4, g_5, g_6$ and g_7 . When solving this optimization problem, these parameters and constraints need to be considered together to find the best combination of parameters to minimize manufacturing costs. How-ever, this is not only an engineering and technical problem but also a comprehensive problem involving economics, mathematical optimization and materials science [25]. The constraint variables are shown in Figure 7.

Its engineering mathematical model is as follows:

Consider $x = [x_1, x_2, x_3, x_4] = [h, l, t, b]$

minimize $f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$

Subject to $g_1(x) = \tau - 13600 \leq 0$,

$$g_2(x) = \sigma - 3000 \leq 0,$$

$$g_3(x) = \delta - 0.025 \leq 0,$$

$$g_4(x) = x_1 - x_4 \leq 0,$$

$$g_5(x) = p - p_c \leq 0,$$

$$g_6(x) = 0.125 - x_1 \leq 0,$$

$$g_7(x) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0,$$

Variable range: $0.1 \leq x_1, x_4 \leq 2.0$, $0.1 \leq x_2, x_3 \leq 10.0$,

$$\tau = \sqrt{\tau_1^2 + 2\tau_1\tau_2\left(\frac{x_2}{2r}\right) + \tau_2^2}, \tau_1 = \frac{p}{x_1x_2\sqrt{2}},$$

$$m = p \left(l + \frac{x_2}{2} \right), j = 2 \left\{ \sqrt{2} x_1 x_2 \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2} \right)^2 \right] \right\},$$

$$r = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2} \right)^2}, \sigma = \frac{6pl}{x_4 x_3^2}, \delta = \frac{6pl^3}{E x_3^2 x_4},$$

$$p_c = \frac{4.013 \sqrt{\frac{x_3^2 x_4^6}{36}}}{l^2} \left(1 - \frac{x_3}{2l} \sqrt{\frac{E}{4G}} \right),$$

$$G = 12 \times 10^6 \text{psi}, E = 30 \times 10^6 \text{psi},$$

$$p = 6000lb, l = 14in, \tau_2 = \frac{mr}{j}$$

The experimental results of each optimization algorithm on the design problem of welded beam are shown in Table 12. We can observe that compared with (HHO [18], GSA [21], SCA [17], BAT [22] and WOA [7] algorithms), the TGES algorithm obtains the most accurate solution.

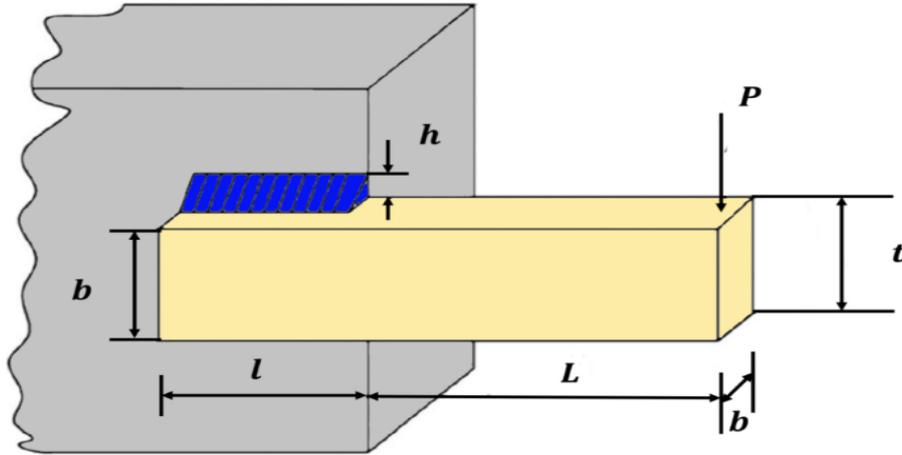


Figure 7: Welded beam design problem.

Table 12: Comparison of result on welded beam design problem.

Algorithm	x_1	x_2	x_3	x_4	Fitness
TGES	0.180404885	2.459756156	9.58442931	0.183027918	1.477565873
HHO	0.136790195	3.435472785	9.611721594	0.182794696	1.544798192
GSA	0.143716387	4.881895534	5.746196063	0.508801521	2.767279004
SCA	0.180527371	3.333711364	7.907088802	0.268758183	1.892191812
BAT	0.714616758	3.38911416	4.810209373	0.726074737	4.833820508
WOA	0.176351548	2.704838541	9.586283198	0.182956253	1.502461029

3.6 Tension/Compression spring design problem

The tension/compression spring problem is a classic structural engineering design problem [26], whose purpose is to minimize the weight of tension/compression spring. To solve the problem, three core variables are needed: wire diameter (d), mean coil diameter (D), and number of active coils (P) [27]. The details of the spring and the three parameters are shown in Figure 8.

The mathematical model of this problem is as follows:

Consider $x = [x_1, x_2, x_3] = [d, D, P]$,

Minimize $f(x) = (x_3 + 2)x_2 x_1^2$,

Subject to $g_1(x) = 1 - \frac{x_2^2 x_3}{71785 x_1^4} \leq 0$,

$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0,$$

$$g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0,$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0,$$

Variable range: $0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.30, 2 \leq x_3 \leq 15$,

On the basis of TGES, the tension/compression spring problem is optimized and the values of relevant parameters are obtained. The optimization results are compared with (HHO [18], GSA [21], SCA [17], BAT [22], WOA [7], MFO [24], SSA [1] and ECO [19] algorithms). The detailed information is shown in Table 13. As can be seen from Table 14, compared with other algorithms, the spring weight obtained by TGES algorithm is 0.01273711. In general, TGES algorithm can effectively obtain the optimal solution of engineering problems and get the best parameter values.

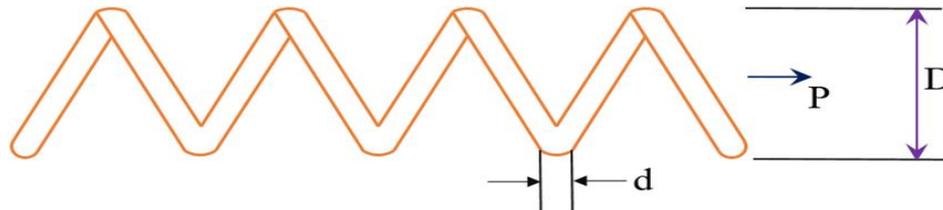


Figure 8: Tension/Compression spring design problem.

Table 13: Comparison of result on tension/compression spring design problem.

Algorithm	S1	S2	S3	Fitness
TGES	0.05	0.317155044	14.06420666	0.01273711
HHO	0.063849134	0.725942211	3.11851335	0.015148022
GSA	0.071000779	1.023526388	5.102182766	0.036645202
SCA	0.061313985	0.635455668	4.601179783	0.01576979
BAT	0.069503226	0.955787433	3.297455882	0.024458994
WOA	0.066135615	0.814084659	2.546423045	0.016188634
MFO	0.05	0.310413658	15	0.01319258
SSA	0.068993951	0.933431788	2	0.017773158
ECO	0.051620974	0.355081963	11.38557575	0.012665375

4.0 CONCLUSION

In this study, an innovative optimization algorithm, namely the Three-Group Exploration Strategy algorithm (TGES), is proposed for solving complex optimization problems, which is inspired by group working tactics. TGES algorithm operates by dividing particles into three distinct groups, each employing a unique update mechanism tailored to accelerate convergence and improve search efficiency.

Through a series of experimental verification in a total of 26 test functions (including unimodal, multimodal and fixed dimensional) benchmark test functions, we comprehensively compared the TGES algorithm with four other optimization algorithms, including PSO, GWO, SCA and SSA algorithms. Experimental results show that the TGES algorithm shows excellent search performance in many characteristics, including convergence speed, solution accuracy, and stability. Additionally, comparative analysis with other algorithms on the CEC-2017 benchmark functions demonstrates the TGES algorithm's high rankings across several test cases. Moreover, the TGES algorithm has shown successful results on applied engineering problems, further highlighting its robustness and effectiveness. It is gratifying that the TGES algorithm has also made remarkable achievements in these practical problems. In summary, this research highlights the TGES algorithm's capability to outperform established optimization techniques, present promising applications for complex optimization problems.

Author Contributions

The sole author, Ayad Ramadhan Ali, conceived, designed, conducted the research, and wrote the manuscript. All aspects of the manuscript, including writing, revising, and final approval, were solely handled by the author.

Availability of Data and Materials

Data sharing is not applicable to this article as no datasets were generated or analyzed during the study.

Declarations

Ethics Approval and Consent to Participate

Ethics approval and consent to participate are not applicable to this study.

Competing interests

The author declares that there are no financial or non-financial competing interests.

Funding

The author received no funding for this study.

Acknowledgments

First, my thanks are due to the staff of the Mathematics Department, College of Science, University of Zakho for their help. Special thanks are also due to my family for their continuous support and many helpful suggestions during my research.

REFERENCE

- [1] J. Xue and B. Shen (2020), "A novel swarm intelligence optimization approach: sparrow search algorithm," *Syst. Sci. Control Eng.*, vol. 8, no. 1, pp. 22–34.
- [2] L. Ke, Q. Zhang, and R. Battiti (2013), "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and antcolony," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1845–1859.
- [3] J. Kennedy and R. Eberhart (1995), "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, ieeep, pp. 1942–1948.
- [4] Y. Zhang (2020), "Coverage optimization and simulation of wireless sensor networks based on particle swarm optimization," *Int. J. Wirel. Inf. Networks*, vol. 27, no. 2, pp. 307–316.
- [5] L. A. I. Yuan-wen and Z. Jie (2021), "Urban bus scheduling optimization based on simulated anneal-adaptive cuckoo search algorithm," *J. Transp. Syst. Eng. Inf. Technol.*, vol. 21, no. 1, p. 183.
- [6] S. Mirjalili, S. M. Mirjalili, and A. Lewis (2014), "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61.
- [7] S. Mirjalili and A. Lewis, "The whale optimization algorithm (2016)," *Adv. Eng. Softw.*, vol. 95, pp. 51–67.
- [8] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman (2020), "Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization," *Eng. Appl. Artif. Intell.*, vol. 90, p. 103541.
- [9] E. Alba and B. Dorronsoro (2005), "The exploration/exploitation tradeoff in dynamic cellular genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 9, no. 2, pp. 126–142.
- [10] D. H. Wolpert and W. G. Macready (1997), "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82.
- [11] A. Layeb, "New hard benchmark functions for global optimization (2022)," *arXiv Prepr. arXiv2202.04606*.
- [12] M. Jamil and X.-S. Yang (2013), "A literature survey of benchmark functions for global optimisation problems," *Int. J. Math. Model. Numer. Optim.*, vol. 4, no. 2, pp. 150–194.
- [13] X.-S. Yang (2010), "Test problems in optimization," *arXiv Prepr. arXiv1008.0549*.
- [14] S.-E. K. Fateen and A. Bonilla-Petriciolet (2014), "Intelligent firefly algorithm for global optimization," *Cuckoo Search Firefly Algorithm Theory Appl.*, Springer, pp. 315–330.
- [15] K. Hussain, M. N. M. Salleh, S. Cheng, and R. Naseem (2017), "Common benchmark functions for metaheuristic evaluation: A review," *JOIV Int. J. Informatics Vis.*, vol. 1, no. 4–2, pp. 218–223.
- [16] G. Wu, R. Mallipeddi, and P. N. Suganthan (2017), "Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization," *Natl. Univ. Def. Technol. Chang. Hunan, PR China Kyungpook Natl. Univ. Daegu, South Korea Nanyang Technol. Univ. Singapore, Tech. Rep.*.
- [17] S. Mirjalili (2016), "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-based Syst.*, vol. 96, pp. 120–133.

- [18] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen (2019), "Harris hawks optimization: Algorithm and applications," *Futur. Gener. Comput. Syst.*, vol. 97, pp. 849–872.
- [19] J. Lian *et al.* (2024), "The educational competition optimizer," *Int. J. Syst. Sci.*, vol. 55, no. 15, pp. 3185–3222.
- [20] B. K. Kannan and S. N. Kramer (1994), "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design," *Journal of Mechanical Design*, vol. 116, pp. 405–411.
- [21] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi (2009), "GSA: a gravitational search algorithm," *Inf. Sci. (Ny).*, vol. 179, no. 13, pp. 2232–2248.
- [22] X. Yang and A. Hossein Gandomi (2012), "Bat algorithm: a novel approach for global engineering optimization," *Eng. Comput.*, vol. 29, no. 5, pp. 464–483.
- [23] T. Ray and P. Saini (2001), "Engineering design optimization using a swarm with an intelligent information sharing among individuals," *Eng. Optim.*, vol. 33, no. 6, pp. 735–748.
- [24] S. Mirjalili (2015), "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-based Syst.*, vol. 89, pp. 228–249.
- [25] C. A. C. Coello (2000), "Use of a self-adaptive penalty approach for engineering optimization problems," *Comput. Ind.*, vol. 41, no. 2, pp. 113–127.
- [26] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi (2013), "Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2592–2612.
- [27] K. Wansasueb, S. Panmanee, N. Panagant, N. Pholdee, S. Bureerat, and A. R. Yildiz (2022), "Hybridised differential evolution and equilibrium optimiser with learning parameters for mechanical and aircraft wing design," *Knowledge-Based Syst.*, vol. 239, p. 107955.