Journal of Network Intelligence ©2024 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **9**, Number **1**, February **2024**

Gannet Optimization Algorithm Enhanced by Quasi-Affine Transformation Algorithm and Its Application in 3D Coverage of Wireless Sensor Network

Yun-Feng Peng

School of Computer Science
Northeast Electric Power University, Jilin 132012, China
pengyunfeng99@163.com

Shu-Chuan Chu, Ru-Yu Wang

College of Computer Science and Engineering
Shandong University of Science and Technology, Qingdao 266590, China
scchu0803@sdust.edu.cn, 202380060003@sdust.edu.cn

Jia Zhao

School of Information Engineering
Nanchang Institute of Technology, Nanchang 330099, China
zhaojia925@163.com

Jeng-Shyang Pan*

College of Computer Science and Engineering
Shandong University of Science and Technology, Qingdao 266590, China
Department of Information Management
Chaoyang University of Technology, Taichung 41349, Taiwan
jengshyangpan@gmail.com

*Corresponding author: Jeng-Shyang Pan

Received September 22, 2023, revised November 9, 2023, accepted December 1, 2023.

Abstract. *The gannet optimization algorithm (GOA) is an effective group intelligence algorithm inspired by the foraging behavior of gannets. Despite its merits, considerable potential exists for enhancing its exploration and convergence capabilities. A gannet optimization algorithm improved by the quasi-affine transformation evolutionary algorithm, restart strategy, and elite selection strategy (QRE-GOA) is proposed in this paper. The algorithm employs the quasi-affine transformation evolutionary algorithm (QUATRE) to matrix the gannet optimization algorithm to search for optimal solutions more accurately. The use of a restart strategy and elite selection can be more effective in preventing the algorithm from falling into local optimal solutions. Comparative experiments were conducted using the CEC2017 benchmark series. A detailed analysis was conducted, we performed a detailed analysis to check the accuracy and convergence speed of the algorithm search in different dimensions. Subsequently, a comprehensive analysis was undertaken to assess the accuracy and convergence speed of the algorithm's search process across various dimensions. The experimental statistical results show that the new algorithm has stronger exploration ability and better convergence ability than the original algorithm. In addition, this paper focuses on process justification so that the experimental results are reliable, trustworthy, and interpretable. Finally, the newly proposed QRE-GOA algorithm is deployed to address the 3D Coverage challenge in Wireless Sensor Networks. Experimental findings unequivocally demonstrate the superior performance of QRE-GOA. These results, underscore its potential for addressing existing limitations in optimization methods.*

Keywords: gannet optimization algorithm, the quasi-affine transformation evolutionary algorithm, restart strategy, elite selection strategy, 3D Coverage.

1. **Introduction.** Swarm intelligence algorithms [1–4], are an optimization algorithm inspired from the life characteristics or behaviors of organisms in nature [5]. The unique feature of these algorithms is the use of a nondifferentiable mechanism that eliminates the need to compute the differentiation of the optimization function, thus simplifying the computational complexity. Due to these properties, it is possible to apply swarm intelligence algorithms to black-box [6] problems where the optimization function cannot be determined (problems where there is no direct knowledge of the internal operations). Due to the versatility and flexibility of these algorithms, they are of practical value in solving various optimization problems as listed below. They are of great interest and popularity among researchers because of their profound impact on production and life. These algorithms can simulate the interaction and cooperation between individuals in a group to achieve group intelligence-like effects and provide new ideas for problem solving. There have been many group intelligence-related algorithms, such as: genetic algorithm (GA) [7], arithmetic optimization algorithm (AOA) [8], particle swarm optimization (PSO) [9], sine cosine algorithm (SCA) [10], gannet optimization algorithm (GOA) [11, 12], differential evolution (DE) [13], phasmatodea population evolution (PPE) [14], tabu search approach (TS) [15], quantum genetic algorithm (QGA) [16], etc.

The focus of this paper is on GOA. GOA is a new meta-heuristic optimization algorithm inspired by the foraging behavior of gannets [17, 18]. In the GOA, the predatory behavior of gannet is simulated and the optimization algorithm is divided into two stages: exploration and exploitation. The algorithm uses four different types of predatory behavior: u-sink, v-sink, sharp turn and random walk. These behavioral patterns allow the algorithm to perform extensive searches in the search space to find potentially optimal solutions. In the search phase, the gannets will be faster after entering the water to develop a strong swimming ability [19, 20]. The algorithm focuses on further potential solutions found to improve the success of predation. By simulating the predation strategy of the gannet, the gannet optimization algorithm gives a new and efficient optimization algorithm suitable for solving various optimization problems. The uniqueness of this algorithm is that it draws inspiration from biological behavior in nature and translates it into a search strategy in the optimization process, providing an interesting and innovative approach to solving practical problems [21–23]. However, the gannet optimization algorithm can easily fall into local optimum problems in high-dimensional and complex problems, so in order to make it have a stronger ability to search for solutions, we added the QUATRE algorithm to enhance its disturbance, so that it can easily jump out of local optimum. So in this paper, the QUATRE is used to improve the gannet optimization problem [24,25]. Before the exploration stage and the exploitation stage, we first use the QUATRE method to matrix the position information of the particles, so that significantly improve the global search ability and local search ability of particles [26].

The improved GOA can better explore the potential solution space and locate the optimal solution more accurately by adding QUATRE, restart strategy and elite selection strategy at the same time. The introduction of the evolutionary matrix increases the diversity of the population, enabling the algorithm to search more extensively during the search process, so that it is more likely to find the global optimal solution. At the same time, the disturbance of the individual's historical optimal position will help to overcome the trap of local optimal solution, which will make the algorithm have stronger local search ability and help to converge to the optimal solution faster in the solution space [27–30].

By combining the evolutionary matrix and the historical optimal position perturbation, the algorithm can perform a more comprehensive and precise search on both global and local levels, making the optimization process more efficient and accurate. Therefore, this improved algorithm has great potential in solving many optimization problems [31,32].

2. Related work.

This section describes the original GOA and explains in detail the need for the inclusion of QUATRE and the hybrid restart strategy and elite selection strategy.

2.1. Gannet Optimization Algorithm.

Gannets, with their stubby, fat bodies and elongated necks, inhabit lakes and coasts around the world in flocks. Their eyes are very sharp, even when flying at high altitude, fish swimming in the water cannot escape the observation of their eyes. Despite being clumsy on land, gannets are remarkably agile in flight and swimming.

The algorithm process of GOA is mainly divided into two stages, namely exploration and exploitation, where in the exploration phase, Gannet's dives are both long and deep U-shaped dives and short and shallow V-shaped dives. The most important part of its core formulation is the position update formulation given in Equation (1).

$$MX_i(t+1) = \begin{cases} X_i(t) + u_1 + u_2, y \geq 0.5 \\ X_i(t) + v_1 + v_2, y < 0.5 \end{cases} \tag{1}$$

Where u_1 is a random number between $-a$ and a, v_1 is a random number between $-b$ and b, and $X_i(t)$ is the $i-th$ individual in the current group.

In the exploitation stage, when the gannets rush into the water in the above two ways, in order to further develop and utilize the resources in the water, two actions need to be taken. The cunning fish in the water often change direction suddenly to avoid the chase of the gannets, which makes the gannets have to expend a lot of energy to catch fast-moving fish. If the gannet possesses sufficient energy, it will exhibit a high capture capacity and successfully catch the fish. In the first scenario, the energy of the gannet dwindles over time, to the point where it may no longer be able to catch the nimble fish. In this case, the gannet performs a Levy motion to search for the next target in a stochastic manner, the core position update formula of which is given in Equation (2). This process allows gannets to search and find effectively in the water when energy is insufficient for targeted capture.

$$MX_i(t+1) = \begin{cases} t \cdot \text{delta} \cdot (X_i(t) - X_{\text{Best}}(t)) + X_i(t), D \geq c \\ X_{\text{Best}}(t) - (X_i(t) - X_{\text{Best}}(t)) \cdot P \cdot t, D < c \end{cases} \tag{2}$$

After many experiments, it is determined that the value of c is 0.2 is the best and $X_{\text{Best}}(t)$ is the best performing individual in the current population. *delta* is given by the equation $delta = D * |X_i(t) - X_{Best(t)}|$ and D is the fishing capacity of the gannet. P is obtained from the levy flight function with the equation $P = Levy(\text{dim})$. The above are only the two core stages of the GOA. But, there is a need for improvement in the GOA algorithm concerning local convergence speed and solution diversity. These enhancements are crucial for boosting the overall performance of the GOA algorithm, enabling it to adapt more effectively to a variety of problems and complexities. For more details, please refer to the paper "Gannet optimization algorithm: A new metaheuristic algorithm for solving engineering optimization problems".

2.2. **QUATRE.** QUATRE draws on the principle of affine transformation in geometric transformation to realize position update [33]. This process is shown by Formula (3).

$$X_{gen+1} = M \otimes X_{gen} + \overline{M} \otimes B \tag{3}$$

$$B = X_{gbest,gen} + F * (X_{r1,gen} - X_{r2,gen}) \tag{4}$$

Among them, X_{gen} is represented as the position coordinate matrix of the first generation particles, which is obtained by performing element multiplication. The co-evolution matrix is composed of matrices B and M. F is generally set at 0.7. $X_{gbest,gen}$ is composed of the best individual position vector in the current population. In addition, $X_{r1,gen}$ and $X_{r2,gen}$ are random matrices generated by randomly perturbing X_{gen}. In addition $\overline{M}$ is to reverse the elements of the M matrix, that is, 0 becomes 1, and 1 becomes 0. The above is the core process of the QUATRE algorithm as follows. As shown in the Formula (6).

$$M_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} = M \tag{5}$$

$$\overline{M} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{6}$$

If there are a total of p particles in the population, the dimension of each particle is *dim*. When $p > dim$, as shown in Formula (7), the first $row * dim$ (*row* is a positive integer) is divided into multiple $dim * dim$ sub-matrices according to the dimension, and the same row and column are performed for each small matrix transformation rules. However, in the process of matrixing the GOA algorithm using the QUATRE algorithm, matrix information will be obtained multiple times, resulting in an increase in the execution time of the algorithm.

$$M_t = \begin{bmatrix} 1 & & \cdots & \\ 1 & 1 & \cdots & \\ & & \cdots & \\ 1 & 1 & \cdots & 1 \\ 1 & & \cdots & \\ 1 & 1 & \cdots & \\ & & \cdots & \\ 1 & 1 & \cdots & 1 \\ 1 & & \cdots & \\ 1 & 1 & \cdots & \\ & & \cdots & \\ 1 & 1 & \cdots & 1 \end{bmatrix} \sim \begin{bmatrix} & 1 & \cdots & \\ & & \cdots & \\ & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ & & \cdots & \\ & & \cdots & \\ & & \cdots & \\ & & \cdots & \\ 1 & 1 & \cdots & \\ 1 & & \cdots & \\ & 1 & \cdots & \\ & & \cdots & \end{bmatrix} = M \tag{7}$$

2.3. **Reboot Strategy and Elite Selection.** The restart strategy, periodically discarding some of the results obtained from the current search, reinitializing some of the searched individuals, and then continuing the search. The purpose of this strategy is to help the GOA jump out of the local optimal solution and improve its global search capability. When the algorithm falls into a local optimal solution, the restart strategy will allow GOA to find a new search direction, so that it is possible to find a better solution. The restart mechanism is designed to enable the algorithm to move as far as possible from a local optimal solution, for which we consider the maximum distance within the search area. We adopted a reasonable approach by computing the diagonal distance of the search area and using it as the particle's escape radius. Specifically, we select a particle in the search space and compute the quarter-diagonal distance in that space. The particles then have the opportunity to hop in all directions with a radius of that distance [34].

The core basic operation process is as follows. First, the RE initialization Formula is (8). Then, calculate the diagonal distance of the search area as shown in Equation (9).

$$RE = Rand \times (u - l) + l \tag{8}$$

$$D = \sqrt{\sum\nolimits_{i=1}^{\dim} (U_i - L_i)^2} \tag{9}$$

where $Rand$ is a random vector with 1 row dim columns, each element is a random number uniformly distributed between 0 and 1. u is the upper bound of the search space, and l is the lower bound of the search space. D is the diagonal distance of the search area calculated by euclidean, U_i is the upper bound matrix of the search space, L_i is the lower bound matrix of the search space.

The third step is to calculate the euclidean distance between RE and the $i-th$ particle and judge whether it is less than a quarter of the search space radius. If it is less than that, reset RE, that is, reset the position information of the $i-th$ particle, as shown in Equation (10). Finally, RE is reset when the restart condition is met as shown in Equation (11).

$$D_1 = \sqrt{\sum\nolimits_{i=1}^{\dim} (RE - X_i)^2} \tag{10}$$

$$RE = RE + (RE - X_i) \times \frac{\sqrt{\sum_i^{\dim} (U - L)^2}}{2000} \tag{11}$$

D_1 is the distance between RE and X, and dim is the dimension of the X matrix. The X matrix stores the position information of the particles, and X_i is the position information of the $i-th$ particle.

The advantage of this method is that its design is relatively simple, but it considers the adaptability of the search area. The escape radius is set based on the characteristics of the search area, which enables the algorithm to adapt to spaces of different shapes and dimensions. This maximum distance-based particle escape method can be applied to various search spaces, making it universal and adaptable. In summary, our restart mechanism aims to improve the global search ability of the algorithm and avoid falling into local optimal solutions. By computing the maximum distance of the search region and applying it to particle escapes, we enable the algorithm to more flexibly explore search spaces of different shapes and dimensions, thus excelling in solving complex problems.

Elite selection retains the best individuals in the current search process in the group and directly passes them on to the next generation. In this way, elite selection ensures that good solutions are retained and not prematurely eliminated. This is very beneficial for maintaining the dominant solution of the population, and also helps to prevent the algorithm from falling into a local optimal solution. Since elite selection retains excellent individuals, the convergence of the entire GOA is improved, and the optimization process reaches a better solution faster. In this way, elite selection plays a very important role in GOA and has a positive impact on the performance and efficiency of the algorithm [35].

The basic operation steps of the elite selection strategy are as follows. First, particles are randomly selected in the particle swarm and σ is initialized, as shown in Equation (12). The second step is to update the c-th particle in the global optimal solution, as shown in Equation (13), the third step checks the boundary of the X matrix, and the fourth step calculates the fitness value of X. Compare the fitness value of X with the global optimal solution. If the fitness value of X is better than the global optimal solution, update the global optimal solution. The pseudocode of the elite selection strategy is as follows Algorithm1.

Algorithm 1 Pseudo-code of the Elite Section Strategy

```
Input: Dim: dimension; gbest: global optimal solution; pbest: individual optimal solution; gbestval: global optimal fitness value;
Output: gbest and pbest;
while stopping condition is not met do
    Step1: randomly select an index Rd from Dim;
    Step2: calculate the variance based on iteration and generation;
    Step3: generate a value Z using Gaussian membership function and gbest(Rd);
    Step4: calculate the new Jump value;
    Step5: create a copy Ju of the current gbest;
    Step6: evaluate the fitness of Ju;
    Step7: update gbest and pbest if Jumpfitness is better than gbestval;
end while
```

$$\sigma = \sigma_u - (\sigma_u - \sigma_l)\frac{It}{M\_It} \tag{12}$$

$$X^c = X^c + (u - l) \times Gaussian(\mu, \sigma^2) \tag{13}$$

Where X^c represents the $c - th$ dimension of X, which is a number of Gaussian distribution, and μ is the mean value and is 0. σ is the standard deviation, it represents the current number of iterations, and M_It is the maximum number of iterations.

When the two strategies of restart strategy and elite selection are used in combination, they can complement each other and further enhance the performance of GOA. The restart strategy increases the exploratory nature of the algorithm, and the elite selection accelerates the convergence of the algorithm, so that GOA can search in the solution space more effectively and achieve better results. Therefore, the combination of these two strategies brings greater advantages to the swarm intelligence algorithm in solving optimization problems, and improves the efficiency and effectiveness of the algorithm. While the restart strategy and elite selection individually contribute significantly to optimizing the GOA algorithm, their combined application is not without potential shortcomings. One potential drawback lies in the complexity introduced by integrating restart strategies and elite selection. Simultaneously using both strategies increases the overall computational burden, especially when dealing with large-scale problems or extensive search spaces. Striking the right balance between global exploration and local exploitation presents challenges, potentially affecting the overall efficiency and effectiveness of the algorithm in solving complex problems.

3. **QRE-GOA.**

Through the analysis of many studies on GOA, we can find that avoiding falling into local extremum is a problem worthy of deep consideration. This paper puts forward several suggestions for improvement on this issue, the specific process is as follows:

First, in the exploitation phase and exploration phase, we introduce the evolution matrix of QUATRE into the initial algorithm, and increase the randomness of particle position updates in the process. This change significantly expands the exploration range of the algorithm. This improvement helps the algorithm to perform a global search in the solution space more effectively, thereby avoiding being limited to the local optimal solution, and significantly improving the global search ability of the algorithm.

Second, in order to further enhance the robustness of the algorithm, we also added a restart strategy and elite selection before each stage. The restart strategy can restart the search process by reinitializing the positions of some particles when the algorithm falls into a local extremum, hoping to find a better solution. The elite selection retains the individuals who performed well in the historical search process, so that these excellent individuals can continue to search the solution space and contribute more opportunities to the global optimization.

Through these improvements, we expect to improve the performance of the GOA in solving complex optimization problems. The introduction of these strategies makes the algorithm more adaptive and global search ability, and can better deal with complex optimization problems such as high-dimensional and nonlinear. The effective combination of these optimization strategies provides a more potential direction for the application and promotion of the GOA. The pseudocode of QRE-GOA is as follows Algorithm 2.

4. **Experiment and Analysis.**

To test the optimization ability of QRE-GOA, we comprehensively compare it with algorithms such as particle swarm optimization(PSO) [36], whale optimization algorithm(WOA) [37,38], antlion optimizer algorithm(AOA) and sine cosine algorithm(SCA) [39]. Since F2 is unstable in CEC2017, we will remove it by default. Table 1 shows the comparison results we obtained on the 29 test functions of CEC2017 [40]. In this table, F1, F3 represent unimodal functions, F4-F10 represent multimodal functions, F11-F20 represent mixed functions, and F21-F30 represent combined functions. Function F2 cannot be tested due to irresistible factors, so we decided to exclude it from the comparison.

Specifically, we used the above 5 algorithms, each algorithm was run 30 times, and 30 particles were used to test on 29 functions, and each function performed 1000 iterations. Finally, we evaluate and compare the results by calculating the mean AVG and standard deviation STD of the errors. In order to show the optimal solution more intuitively, we

Algorithm 2 Pseudo-code of the QRE-GOA

```
Input: $Dim$: dimension; $N$: population size; $M\_It$: total number of iterations;
Output: The fitness value error and position of the gannet;
Initialize the population $X$, $y$ is a random number between 0 and 1;
Initializing the memory matrix $MX$;
while stopping condition is not met do
    Initialize a matrix with N rows and Dim columns; $K = N \div Dim$;
    for $y = 0{:}K - 1$ do
        making it a lower triangular matrix $M_t$;
    end for
    Process the rest and become the lower triangular matrix $M_t$;
    Randomly swap each row of $M$;
    for $t = 1{:}N$ do
        Randomly arrange the columns of each row in the label;
    end for
    Randomly swap $X$ rows to get $X_{r1}$ and $X_{r2}$;
    Updates position $X$ using Equation (4) and Equation (3);
    Elite Section Strategy
    Restart Strategy:
    Step1: randomly select an index $j$ from $p$;
    Step2: apply Restart operation to the selected position;
    Step3: apply space bound operation to the selected position;
    if $rand > 0.5$ then
        for $MX_i$ do
            Update $MX_i$ using Equation (1);
            Compare $MX_i$ and $X_i$ to determine whether to update $X_i$;
        end for
    else
        for $MX_i$ do
            Update $MX_i$ using Equation (1);
            Compare $MX_i$ and $X_i$ to determine whether to update $X_i$;
        end for
    end if
end while
```

use bold font to mark in the table. The research in this paper focuses on the minimization of the problem.

4.1. **Comparison of QRE-GOA with other algorithms on 29 test functions.** Through the running results in Table 1, we can see the comparison between QRE-GOA and other algorithms, and record the number of wins of QRE-GOA when compared with other algorithms in a marked way. Among them, not only the old algorithm PSO is compared, but also three relatively new algorithms WOA, AOA, and SCA, which can further prove the authority and transparency of the QRE-GOA. And it can be clearly seen that the superiority of QRE-GOA compared with other algorithms proves that the QRE-GOA is particularly good, because the average error of the fitness value of the QRE-GOA compared with other algorithms wins at 28 At the same time, the number of wins of the standard deviation of the fitness value compared with other algorithms is more than 28 except for SCA.

In Table 1, we can clearly understand the performance of QRE-GOA on 29 different functions, as well as the competitive results with other optimization algorithms. These comparative results will help to evaluate the performance and scope of application of QRE-GOA, and provide a valuable reference for further research and application.

Table 1. 10 Dim Simulation Results of CEC 2017 Benchmark Function.

Algorithm	Measure	F1	F3	F4	F5	F6	F7	F8	F9	F10	F11
PSO	AVG	6.24×10^{08}	1.08×10^{03}	5.78×10^{01}	3.30×10^{01}	1.68×10^{01}	3.88×10^{01}	2.59×10^{01}	1.33×10^{02}	1.04×10^{03}	1.18×10^{02}
	STD	1.13×10^{09}	4.57×10^{03}	7.19×10^{01}	1.41×10^{01}	1.12×10^{01}	1.45×10^{01}	9.94×10^{00}	1.67×10^{02}	3.94×10^{02}	9.31×10^{01}
WOA	AVG	2.67×10^{07}	2.49×10^{03}	4.78×10^{01}	4.79×10^{01}	3.72×10^{01}	8.77×10^{01}	4.45×10^{01}	4.52×10^{02}	1.06×10^{03}	1.18×10^{02}
	STD	9.03×10^{07}	2.37×10^{03}	4.53×10^{01}	2.18×10^{01}	1.17×10^{01}	2.37×10^{01}	1.48×10^{01}	2.80×10^{02}	3.32×10^{02}	7.94×10^{01}
AOA	AVG	7.80×10^{09}	1.06×10^{04}	4.82×10^{02}	5.77×10^{01}	3.87×10^{01}	9.65×10^{01}	3.07×10^{01}	5.23×10^{02}	1.10×10^{03}	1.42×10^{03}
	STD	3.40×10^{09}	2.74×10^{03}	3.15×10^{02}	1.81×10^{01}	9.35×10^{00}	1.58×10^{01}	7.42×10^{00}	2.12×10^{02}	2.50×10^{02}	1.66×10^{03}
SCA	AVG	8.88×10^{08}	1.49×10^{03}	6.01×10^{01}	4.89×10^{01}	1.97×10^{01}	7.84×10^{01}	4.02×10^{01}	1.19×10^{02}	1.29×10^{03}	1.27×10^{02}
	STD	2.95×10^{08}	7.60×10^{02}	3.10×10^{01}	6.36×10^{00}	3.77×10^{00}	1.04×10^{01}	7.83×10^{00}	7.85×10^{01}	**2.15×10^{02}**	5.95×10^{01}
QRE-GOA	AVG	**6.52×10^{-12}**	**9.47×10^{-15}**	**5.32×10^{-05}**	**1.23×10^{01}**	**6.22×10^{-05}**	**2.28×10^{01}**	**1.33×10^{01}**	**1.04×10^{-08}**	**4.85×10^{02}**	**6.07×10^{00}**
	STD	**1.30×10^{-11}**	**1.02×10^{-14}**	**1.17×10^{-04}**	**6.08×10^{00}**	**9.95×10^{-05}**	**5.82×10^{00}**	**6.13×10^{00}**	**5.69×10^{-08}**	2.86×10^{02}	**4.48×10^{00}**
		F12	F13	F14	F15	F16	F17	F18	F19	F20	F21
PSO	AVG	1.13×10^{06}	2.32×10^{03}	1.00×10^{03}	1.10×10^{03}	2.30×10^{02}	1.12×10^{02}	1.84×10^{04}	6.30×10^{03}	9.07×10^{01}	2.26×10^{02}
	STD	2.90×10^{06}	6.81×10^{03}	4.97×10^{03}	2.50×10^{03}	1.49×10^{02}	6.44×10^{01}	1.67×10^{04}	1.26×10^{04}	5.37×10^{01}	3.51×10^{01}
WOA	AVG	5.40×10^{06}	1.13×10^{04}	9.10×10^{02}	9.33×10^{03}	3.46×10^{02}	9.39×10^{01}	1.48×10^{04}	2.69×10^{04}	2.14×10^{02}	2.47×10^{02}
	STD	4.83×10^{06}	8.62×10^{03}	1.32×10^{03}	8.15×10^{03}	1.53×10^{02}	3.97×10^{01}	1.25×10^{04}	3.59×10^{04}	7.00×10^{01}	4.06×10^{01}
AOA	AVG	3.37×10^{07}	1.37×10^{04}	8.21×10^{03}	1.46×10^{04}	4.86×10^{02}	2.13×10^{02}	1.58×10^{04}	4.61×10^{04}	1.59×10^{02}	2.27×10^{02}
	STD	7.94×10^{07}	1.06×10^{04}	8.52×10^{03}	5.45×10^{03}	1.81×10^{02}	1.18×10^{02}	7.46×10^{03}	3.62×10^{04}	7.33×10^{01}	**2.93×10^{01}**
SCA	AVG	1.90×10^{07}	3.00×10^{04}	4.02×10^{02}	1.06×10^{03}	1.60×10^{02}	7.92×10^{01}	2.05×10^{05}	4.74×10^{03}	1.05×10^{02}	**1.53×10^{02}**
	STD	1.38×10^{07}	2.10×10^{04}	2.70×10^{02}	1.01×10^{03}	8.11×10^{01}	**1.36×10^{01}**	1.25×10^{05}	5.10×10^{03}	3.16×10^{01}	5.93×10^{01}
QRE-GOA	AVG	**2.47×10^{02}**	**9.31×10^{00}**	**1.28×10^{01}**	**3.35×10^{00}**	**1.07×10^{02}**	**3.40×10^{01}**	**1.55×10^{01}**	**2.09×10^{00}**	**1.09×10^{01}**	1.82×10^{02}
	STD	**1.36×10^{02}**	**6.62×10^{00}**	**1.01×10^{01}**	**2.90×10^{00}**	**1.05×10^{02}**	4.22×10^{01}	**9.68×10^{00}**	**2.04×10^{00}**	**1.11×10^{01}**	5.17×10^{01}
		F22	F23	F24	F25	F26	F27	F28	F29	F30	WIN
PSO	AVG	2.40×10^{02}	3.45×10^{02}	3.69×10^{02}	4.54×10^{02}	8.54×10^{02}	4.27×10^{02}	6.10×10^{02}	3.54×10^{02}	9.78×10^{05}	29
	STD	2.86×10^{02}	1.87×10^{01}	6.14×10^{01}	4.29×10^{01}	5.05×10^{02}	2.74×10^{01}	1.16×10^{02}	7.44×10^{01}	1.58×10^{06}	29
WOA	AVG	2.95×10^{02}	3.46×10^{02}	3.68×10^{02}	4.50×10^{02}	8.97×10^{02}	4.38×10^{02}	6.65×10^{02}	4.97×10^{02}	1.00×10^{06}	29
	STD	3.75×10^{02}	2.04×10^{01}	6.21×10^{01}	2.68×10^{01}	5.13×10^{02}	4.31×10^{01}	1.32×10^{02}	1.10×10^{02}	7.99×10^{05}	29
AOA	AVG	7.73×10^{02}	4.32×10^{02}	4.35×10^{02}	7.12×10^{02}	1.34×10^{03}	5.37×10^{02}	9.45×10^{02}	5.31×10^{02}	2.41×10^{07}	29
	STD	2.86×10^{02}	2.95×10^{01}	4.57×10^{01}	1.30×10^{02}	3.75×10^{02}	5.51×10^{01}	1.41×10^{02}	1.59×10^{02}	2.82×10^{07}	28
SCA	AVG	1.73×10^{02}	3.59×10^{02}	3.81×10^{02}	4.71×10^{02}	5.03×10^{02}	4.05×10^{02}	4.98×10^{02}	3.42×10^{02}	1.38×10^{06}	28
	STD	2.79×10^{01}	5.93×10^{00}	**2.69×10^{01}**	2.44×10^{01}	**6.18×10^{01}**	**2.04×10^{00}**	**8.24×10^{01}**	4.86×10^{01}	9.60×10^{05}	23
QRE-GOA	AVG	**9.56×10^{01}**	**3.12×10^{02}**	**3.29×10^{02}**	**4.20×10^{02}**	**3.97×10^{02}**	**3.99×10^{02}**	**4.48×10^{02}**	**2.74×10^{02}**	**2.74×10^{05}**	/
	STD	**1.94×10^{01}**	**5.47×10^{00}**	6.25×10^{01}	**$2.35E \times 10^{01}$**	2.85×10^{02}	1.12×10^{01}	1.69×10^{02}	**3.66×10^{01}**	**3.91×10^{05}**	/

4.2. **Comparison of QRE-GOA and its different components on 29 test functions.** At the same time, in order to prove that the GOA (QRE-GOA) after matrixing, restart strategy and elite selection is superior to the original GOA and its components, we tested them on 29 test functions in CEC2017, and the results are shown in Table 2 Show. Compared with the original algorithm GOA, QRE-GOA has 27 wins compared with the average value of its fitness value error, and 26 wins compared with the standard deviation of its fitness value error. In addition, after further matrixing with QUATRE and after restarting the optimized algorithm, the number of wins of the mean value of the fitness value error and the standard deviation of the fitness value error exceeded 20 times.

GOA with elite selection. The results show that the optimization strategy of each part in QRE-GOA is effective and necessary. Through the data display in Table 2, we clearly present the impact of these optimization strategies on the performance of the algorithm.

These test results confirm the effectiveness of the QRE-GOA for us, and further verify the contribution of matrix partition, restart strategy and elite selection to the algorithm performance. This has important implications for gaining insight into the benefits of QRE-GOA and the impact of optimization strategies. Therefore, the results in Table 2 provide solid empirical support for our research and provide a useful reference for future algorithm improvement and application. AVG represents the average value of fitness error, STD represents the standard deviation of fitness error. Bold values represent the smallest fitness error. QGOA represents an improvement to the GOA algorithm, which only enhances the perturbation characteristics of the original algorithm by introducing the QUATRE element. QRGOA shows that more comprehensive improvement measures have been taken in the GOA algorithm. In addition to the introduction of QUATRE, a restart strategy is also introduced, which means that the algorithm will be restarted periodically during the iteration process to explore the search space more actively.

Table 2. Comparison between QRE-GOA algorithm and several improved GOA.

Algorithm	Measure	F1	F3	F4	F5	F6	F7	F8	F9	F10	F11
GOA	AVG	2.58×10^{03}	6.49×10^{-05}	3.20×10^{00}	1.88×10^{01}	1.42×10^{-01}	2.38×10^{01}	1.60×10^{01}	3.30×10^{00}	7.45×10^{02}	1.70×10^{01}
	STD	2.70×10^{03}	7.84×10^{-05}	1.45×10^{00}	8.79×10^{00}	4.39×10^{-01}	7.55×10^{00}	7.88×10^{00}	1.60×10^{01}	2.87×10^{02}	3.39×10^{01}
QGOA	AVG	$\mathbf{1.89 \times 10^{-15}}$	$\mathbf{3.79 \times 10^{-15}}$	$\mathbf{1.34 \times 10^{-09}}$	1.72×10^{01}	1.28×10^{-02}	2.70×10^{01}	1.95×10^{01}	2.75×10^{-01}	7.44×10^{02}	9.73×10^{00}
	STD	$\mathbf{4.91 \times 10^{-15}}$	2.09×10^{-14}	$\mathbf{7.02 \times 10^{-09}}$	7.02×10^{00}	4.06×10^{-02}	6.90×10^{00}	8.27×10^{00}	4.26×10^{-01}	3.36×10^{02}	6.98×10^{00}
QRGOA	AVG	4.07×10^{-13}	3.79×10^{-15}	1.78×10^{-05}	1.23×10^{01}	6.52×10^{-05}	2.49×10^{01}	1.51×10^{01}	1.51×10^{-02}	$\mathbf{4.49 \times 10^{02}}$	$\mathbf{5.50 \times 10^{00}}$
	STD	2.11×10^{-12}	1.44×10^{-14}	3.40×10^{-05}	6.80×10^{00}	1.54×10^{-04}	6.70×10^{00}	6.59×10^{00}	8.29×10^{-02}	$\mathbf{2.64 \times 10^{02}}$	$\mathbf{3.20 \times 10^{00}}$
QRE-GOA	AVG	6.52×10^{-12}	9.47×10^{-15}	5.32×10^{-05}	$\mathbf{1.23 \times 10^{01}}$	$\mathbf{6.22 \times 10^{-05}}$	$\mathbf{2.28 \times 10^{01}}$	$\mathbf{1.33 \times 10^{01}}$	$\mathbf{1.04 \times 10^{-08}}$	4.85×10^{02}	6.07×10^{00}
	STD	1.30×10^{-11}	$\mathbf{1.02 \times 10^{-14}}$	1.17×10^{-04}	$\mathbf{6.08 \times 10^{00}}$	$\mathbf{9.95 \times 10^{-05}}$	$\mathbf{5.82 \times 10^{00}}$	$\mathbf{6.13 \times 10^{00}}$	$\mathbf{5.69 \times 10^{-08}}$	2.86×10^{02}	4.48×10^{00}
		F12	F13	F14	F15	F16	F17	F18	F19	F20	F21
GOA	AVG	2.02×10^{04}	4.84×10^{03}	4.88×10^{01}	1.21×10^{02}	$\mathbf{5.21 \times 10^{01}}$	3.96×10^{01}	8.55×10^{03}	6.94×10^{01}	4.98×10^{01}	1.83×10^{02}
	STD	1.88×10^{04}	6.95×10^{03}	2.70×10^{01}	9.17×10^{01}	$\mathbf{7.62 \times 10^{01}}$	$\mathbf{2.94 \times 10^{01}}$	6.78×10^{03}	1.05×10^{02}	5.61×10^{01}	5.39×10^{01}
QGOA	AVG	$\mathbf{2.44 \times 10^{02}}$	1.09×10^{01}	3.51×10^{02}	4.15×10^{00}	1.56×10^{02}	4.47×10^{01}	1.83×10^{01}	2.84×10^{00}	3.69×10^{01}	1.96×10^{02}
	STD	$\mathbf{1.31 \times 10^{02}}$	4.74×10^{00}	1.75×10^{03}	3.78×10^{00}	1.35×10^{02}	5.00×10^{01}	1.13×10^{01}	3.73×10^{00}	4.52×10^{01}	4.93×10^{01}
QRGOA	AVG	9.25×10^{02}	9.34×10^{00}	1.47×10^{01}	3.36×10^{00}	1.09×10^{02}	3.93×10^{01}	1.34×10^{01}	$\mathbf{1.91 \times 10^{00}}$	1.70×10^{01}	1.99×10^{02}
	STD	1.22×10^{03}	$\mathbf{4.01 \times 10^{00}}$	1.10×10^{01}	2.91×10^{00}	1.43×10^{02}	4.92×10^{01}	1.09×10^{01}	$\mathbf{1.33 \times 10^{00}}$	2.80×10^{01}	3.97×10^{01}
QRE-GOA	AVG	2.47×10^{02}	$\mathbf{9.31 \times 10^{00}}$	$\mathbf{1.28 \times 10^{01}}$	$\mathbf{3.35 \times 10^{00}}$	1.07×10^{02}	$\mathbf{3.40 \times 10^{01}}$	$\mathbf{1.34 \times 10^{01}}$	2.09×10^{00}	$\mathbf{1.09 \times 10^{01}}$	$\mathbf{1.82 \times 10^{02}}$
	STD	1.36×10^{02}	6.62×10^{00}	$\mathbf{1.01 \times 10^{01}}$	$\mathbf{2.90 \times 10^{00}}$	1.05×10^{02}	4.22×10^{01}	$\mathbf{9.68 \times 10^{00}}$	2.04×10^{00}	$\mathbf{1.11 \times 10^{01}}$	$\mathbf{3.97 \times 10^{01}}$
		F22	F23	F24	F25	F26	F27	F28	F29	F30	WIN
GOA	AVG	9.56×10^{01}	3.18×10^{02}	3.35×10^{02}	4.32×10^{02}	4.93×10^{02}	4.00×10^{02}	5.37×10^{02}	2.91×10^{02}	$\mathbf{1.84 \times 10^{05}}$	27
	STD	2.09×10^{01}	7.59×10^{00}	6.44×10^{01}	$\mathbf{2.08 \times 10^{01}}$	3.50×10^{02}	1.42×10^{01}	1.67×10^{02}	4.86×10^{01}	$\mathbf{3.67 \times 10^{05}}$	26
QGOA	AVG	9.58×10^{01}	3.16×10^{02}	3.39×10^{02}	4.30×10^{02}	$\mathbf{3.54 \times 10^{02}}$	4.01×10^{02}	$\mathbf{4.18 \times 10^{02}}$	3.06×10^{02}	2.04×10^{05}	23
	STD	2.34×10^{01}	8.38×10^{00}	4.57×10^{01}	2.22×10^{01}	$\mathbf{2.68 \times 10^{02}}$	1.48×10^{01}	1.67×10^{02}	6.30×10^{01}	4.72×10^{05}	22
QRGOA	AVG	9.93×10^{01}	3.12×10^{02}	3.36×10^{02}	4.20×10^{02}	4.25×10^{02}	3.99×10^{02}	4.61×10^{02}	2.86×10^{02}	3.16×10^{05}	23
	STD	$\mathbf{1.29 \times 10^{01}}$	5.95×10^{00}	$\mathbf{4.55 \times 10^{01}}$	2.35×10^{01}	3.50×10^{02}	1.12×10^{01}	$\mathbf{1.45 \times 10^{02}}$	3.77×10^{01}	4.68×10^{05}	20
QRE-GOA	AVG	$\mathbf{9.56 \times 10^{01}}$	$\mathbf{3.12 \times 10^{02}}$	$\mathbf{3.29 \times 10^{02}}$	$\mathbf{4.20 \times 10^{02}}$	3.97×10^{02}	$\mathbf{3.96 \times 10^{02}}$	4.48×10^{02}	$\mathbf{2.74 \times 10^{02}}$	2.74×10^{05}	/
	STD	1.94×10^{01}	$\mathbf{5.47 \times 10^{00}}$	6.25×10^{01}	2.35×10^{01}	2.85×10^{02}	$\mathbf{3.49 \times 10^{00}}$	1.67×10^{02}	$\mathbf{3.66 \times 10^{01}}$	3.91×10^{05}	/

4.3. Different dimensions of QRE-GOA compared on 29 test functions.

In order to comprehensively prove that QRE-GOA outperforms the GOA algorithm in all dimensions, this study evaluates its performance on the CEC2017 test function in detail. The QRE-GOA algorithm introduces innovative elements such as evolutionary matrix, restart strategy, and elite selection, and these improvements bring significant advantages.

First, the introduction of the evolution matrix endows QRE-GOA with stronger search capabilities. By referring to the evolution matrix, the algorithm can explore in the solution space more flexibly, so as to better capture the characteristics of the objective function. This strategy makes QRE-GOA not only have a very good search ability in low dimensions, but also has a good search ability in high latitudes.

Second, the restart strategy and elite selection mechanism of QRE-GOA further enhance the global search ability of the algorithm. The restart strategy enables the algorithm to jump out of the local optimal solution and re-explore the solution space, which helps to find a better solution. The elite selection mechanism can retain the current best solution, so as to ensure that the algorithm will not lose the advantages it has gained during the optimization process. These properties are more important in multi-dimensional problems, because the search in high-dimensional spaces is more difficult and requires stronger global search capabilities.

On the whole, the advantages of QRE-GOA in different dimensions are mainly reflected in stronger search ability, more efficient global search and better local search jump-out ability. These advantages enable QRE-GOA to better deal with complex, high-dimensional optimization problems, and show better performance than traditional GOA algorithms. Through comparative experiments on the CEC2017 test function, we demonstrate the excellent performance of QRE-GOA in different dimensions, which provides strong support for its broad potential in practical applications. From Table 3, we can clearly see that QRE-GOA, compared with the original algorithm GOA in 10 and 30 dimensions, has the average number of wins of the fitness value error as high as 27 times and the standard deviation of the fitness value error The number of wins is as high as 25, and the number of wins compared with the average value of the fitness value error and the standard deviation of the fitness value error on the 50-dimension has reached more than 19 times, which further shows that QRE-GOA has different latitudes The optimization of GOA has achieved remarkable results.

Table 3. Comparison of optimization rates between QRE-GOA and GOA in different dimensions.

Dimension	Algorithm	Measure	F1	F3	F4	F5	F6	F7	F8	F9
D=10	GOA	AVG	2.58×10^{03}	6.49×10^{-05}	3.20×10^{00}	1.88×10^{01}	1.42×10^{-01}	2.38×10^{01}	1.60×10^{01}	3.30×10^{00}
		STD	2.70×10^{03}	7.84×10^{-05}	1.45×10^{00}	8.79×10^{00}	4.39×10^{-01}	7.55×10^{00}	7.88×10^{00}	1.60×10^{01}
	QRE-GOA	AVG	**6.52×10^{-12}**	**9.47×10^{-15}**	**5.32×10^{-05}**	**1.23×10^{01}**	**6.22×10^{-05}**	**2.28×10^{01}**	**1.33×10^{01}**	**1.04×10^{-08}**
		STD	**1.30×10^{-11}**	**1.02×10^{-14}**	**1.17×10^{-04}**	**6.08×10^{00}**	**9.95×10^{-05}**	**5.82×10^{00}**	**6.13×10^{00}**	**5.69×10^{-08}**
D=30	GOA	AVG	4.04×10^{03}	2.29×10^{04}	9.36×10^{01}	1.26×10^{02}	1.29×10^{01}	1.88×10^{02}	1.12×10^{02}	1.46×10^{03}
		STD	4.53×10^{03}	9.26×10^{03}	2.92×10^{01}	3.26×10^{01}	6.23×10^{00}	4.32×10^{01}	**2.56×10^{01}**	7.05×10^{02}
	QRE-GOA	AVG	**3.31×10^{01}**	**1.61×10^{04}**	**8.05×10^{01}**	**7.59×10^{01}**	**1.63×10^{00}**	**1.22×10^{02}**	**9.01×10^{01}**	**3.13×10^{02}**
		STD	**7.16×10^{01}**	**5.91×10^{03}**	**2.46×10^{01}**	**2.09×10^{01}**	**1.93×10^{00}**	**2.30×10^{01}**	2.75×10^{01}	**3.00×10^{02}**
D=50	GOA	AVG	1.78×10^{04}	**1.04×10^{05}**	1.71×10^{02}	**2.61×10^{02}**	3.01×10^{01}	4.37×10^{02}	**2.70×10^{02}**	7.71×10^{03}
		STD	9.48×10^{03}	**2.26×10^{04}**	5.37×10^{01}	5.02×10^{01}	1.06×10^{01}	8.43×10^{01}	4.67×10^{01}	2.54×10^{03}
	QRE-GOA	AVG	**6.32×10^{03}**	2.27×10^{05}	**1.61×10^{02}**	2.92×10^{02}	**2.10×10^{-02}**	**3.88×10^{02}**	2.96×10^{02}	**1.37×10^{02}**
		STD	**7.47×10^{03}**	3.99×10^{04}	**5.25×10^{01}**	**3.41×10^{01}**	**2.29×10^{-02}**	**2.66×10^{01}**	**2.82×10^{01}**	**2.04×10^{02}**
			F10	F11	F12	F13	F14	F15	F16	F17
D=10	GOA	AVG	7.45×10^{02}	1.70×10^{01}	2.02×10^{04}	4.84×10^{03}	4.88×10^{01}	1.21×10^{02}	**5.21×10^{01}**	3.96×10^{01}
		STD	2.87×10^{02}	3.39×10^{01}	1.88×10^{04}	6.95×10^{03}	2.70×10^{01}	9.17×10^{01}	**7.62×10^{01}**	**2.94×10^{01}**
	QRE-GOA	AVG	**4.85×10^{02}**	**6.07×10^{00}**	**2.47×10^{02}**	**9.31×10^{00}**	**1.28×10^{01}**	**3.35×10^{00}**	1.07×10^{02}	**3.40×10^{01}**
		STD	**2.86×10^{02}**	**4.48×10^{00}**	**1.36×10^{02}**	**6.62×10^{00}**	**1.01×10^{01}**	**2.90×10^{00}**	1.05×10^{02}	4.22×10^{01}
D=30	GOA	AVG	4.24×10^{03}	1.20×10^{02}	1.81×10^{06}	2.50×10^{04}	2.16×10^{04}	**3.50×10^{03}**	1.02×10^{03}	**3.88×10^{02}**
		STD	8.02×10^{02}	**3.94×10^{01}**	1.42×10^{06}	1.88×10^{04}	2.09×10^{04}	**3.04×10^{03}**	2.73×10^{02}	**1.73×10^{02}**
	QRE-GOA	AVG	**3.08×10^{03}**	**8.78×10^{01}**	**6.21×10^{04}**	**1.13×10^{04}**	**1.59×10^{03}**	6.99×10^{03}	**8.60×10^{02}**	4.15×10^{02}
		STD	**6.65×10^{02}**	4.42×10^{01}	**3.75×10^{04}**	**1.73×10^{04}**	**1.61×10^{03}**	8.82×10^{03}	**2.68×10^{02}**	2.08×10^{02}
D=50	GOA	AVG	**7.22×10^{03}**	2.58×10^{02}	1.47×10^{07}	1.80×10^{04}	**1.76×10^{05}**	1.59×10^{04}	**1.81×10^{03}**	1.22×10^{03}
		STD	1.59×10^{03}	7.77×10^{01}	7.18×10^{06}	1.22×10^{04}	**1.56×10^{05}**	9.61×10^{03}	4.40×10^{02}	2.74×10^{02}
	QRE-GOA	AVG	1.04×10^{04}	**2.71×10^{02}**	**7.29×10^{06}**	**6.68×10^{03}**	3.97×10^{05}	**6.00×10^{03}**	2.10×10^{03}	**1.16×10^{03}**
		STD	**7.68×10^{02}**	**6.11×10^{01}**	**3.35×10^{06}**	**7.82×10^{03}**	3.14×10^{05}	**6.68×10^{03}**	**3.62×10^{02}**	**2.35×10^{02}**
			F18	F19	F20	F21	F22	F23	F24	F25
D=10	GOA	AVG	8.55×10^{03}	6.94×10^{01}	4.98×10^{01}	1.83×10^{02}	9.56×10^{01}	3.18×10^{02}	3.35×10^{02}	4.32×10^{02}
		STD	6.78×10^{03}	1.05×10^{02}	5.61×10^{01}	5.39×10^{01}	2.09×10^{01}	7.59×10^{00}	6.44×10^{01}	**2.08×10^{01}**
	QRE-GOA	AVG	**1.34×10^{01}**	**2.09×10^{00}**	**1.09×10^{01}**	**1.82×10^{02}**	**9.56×10^{01}**	**3.12×10^{02}**	**3.29×10^{02}**	**4.20×10^{02}**
		STD	**9.68×10^{00}**	**2.04×10^{00}**	**1.11×10^{01}**	**3.97×10^{01}**	**1.94×10^{01}**	**5.47×10^{00}**	**6.25×10^{01}**	2.35×10^{01}
D=30	GOA	AVG	1.89×10^{05}	9.99×10^{03}	4.04×10^{02}	3.18×10^{02}	**2.15×10^{03}**	4.99×10^{02}	5.78×10^{02}	4.02×10^{02}
		STD	1.59×10^{05}	1.02×10^{04}	1.94×10^{02}	3.05×10^{01}	2.32×10^{03}	5.60×10^{01}	5.54×10^{01}	1.77×10^{01}
	QRE-GOA	AVG	**1.13×10^{05}**	**1.40×10^{03}**	**3.73×10^{02}**	**2.89×10^{02}**	2.36×10^{03}	**4.25×10^{02}**	**4.91×10^{02}**	**3.91×10^{02}**
		STD	**1.49×10^{05}**	**2.81×10^{03}**	**1.56×10^{02}**	**2.64×10^{01}**	**1.71×10^{03}**	**2.20×10^{01}**	**2.50×10^{01}**	**1.35×10^{01}**
D=50	GOA	AVG	**1.44×10^{06}**	2.01×10^{04}	**9.71×10^{02}**	**4.65×10^{02}**	**7.85×10^{03}**	7.69×10^{02}	8.54×10^{02}	6.03×10^{02}
		STD	**1.35×10^{06}**	**1.18×10^{04}**	3.43×10^{02}	5.89×10^{01}	2.51×10^{03}	8.77×10^{01}	**1.25×10^{02}**	**2.37×10^{01}**
	QRE-GOA	AVG	4.09×10^{06}	**1.00×10^{04}**	1.08×10^{03}	4.94×10^{02}	1.04×10^{04}	**7.35×10^{02}**	**8.12×10^{02}**	**5.60×10^{02}**
		STD	2.93×10^{06}	1.19×10^{04}	**2.86×10^{02}**	**3.86×10^{01}**	**2.12×10^{03}**	**3.51×10^{01}**	2.45×10^{01}	2.86×10^{01}
			F26	F27	F28	F29	F30		WIN	
D=10	GOA	AVG	4.93×10^{02}	4.00×10^{02}	5.37×10^{02}	2.91×10^{02}	**1.84×10^{05}**	D=10	AVG: 27	
		STD	3.50×10^{02}	1.42×10^{01}	1.67×10^{02}	4.86×10^{01}	**3.67×10^{05}**			
	QRE-GOA	AVG	**3.97×10^{02}**	**3.96×10^{02}**	**4.48×10^{02}**	**2.74×10^{02}**	2.74×10^{05}		STD: 25	
		STD	**2.85×10^{02}**	**3.49×10^{00}**	**1.67×10^{02}**	**3.66×10^{01}**	3.91×10^{05}			
D=30	GOA	AVG	2.83×10^{03}	5.47×10^{02}	4.35×10^{02}	1.01×10^{03}	1.28×10^{04}	D=30	AVG: 27	
		STD	1.22×10^{03}	2.42×10^{01}	**2.22×10^{01}**	2.47×10^{02}	6.92×10^{03}			
	QRE-GOA	AVG	**1.92×10^{03}**	**5.30×10^{02}**	**4.02×10^{02}**	**7.92×10^{02}**	**6.56×10^{03}**		STD: 25	
		STD	**5.77×10^{02}**	**1.46×10^{01}**	4.57×10^{01}	**1.85×10^{02}**	**3.78×10^{03}**			
D=50	GOA	AVG	4.86×10^{03}	8.80×10^{02}	5.76×10^{02}	1.56×10^{03}	**1.40×10^{06}**	D=50	AVG: 19	
		STD	2.33×10^{03}	1.17×10^{02}	3.08×10^{01}	3.16×10^{02}	**5.13×10^{05}**			
	QRE-GOA	AVG	**3.86×10^{03}**	**6.30×10^{02}**	**5.17×10^{02}**	**1.28×10^{03}**	2.18×10^{06}		STD: 22	
		STD	**4.54×10^{02}**	**5.41×10^{01}**	**2.30×10^{01}**	**2.85×10^{02}**	1.16×10^{06}			

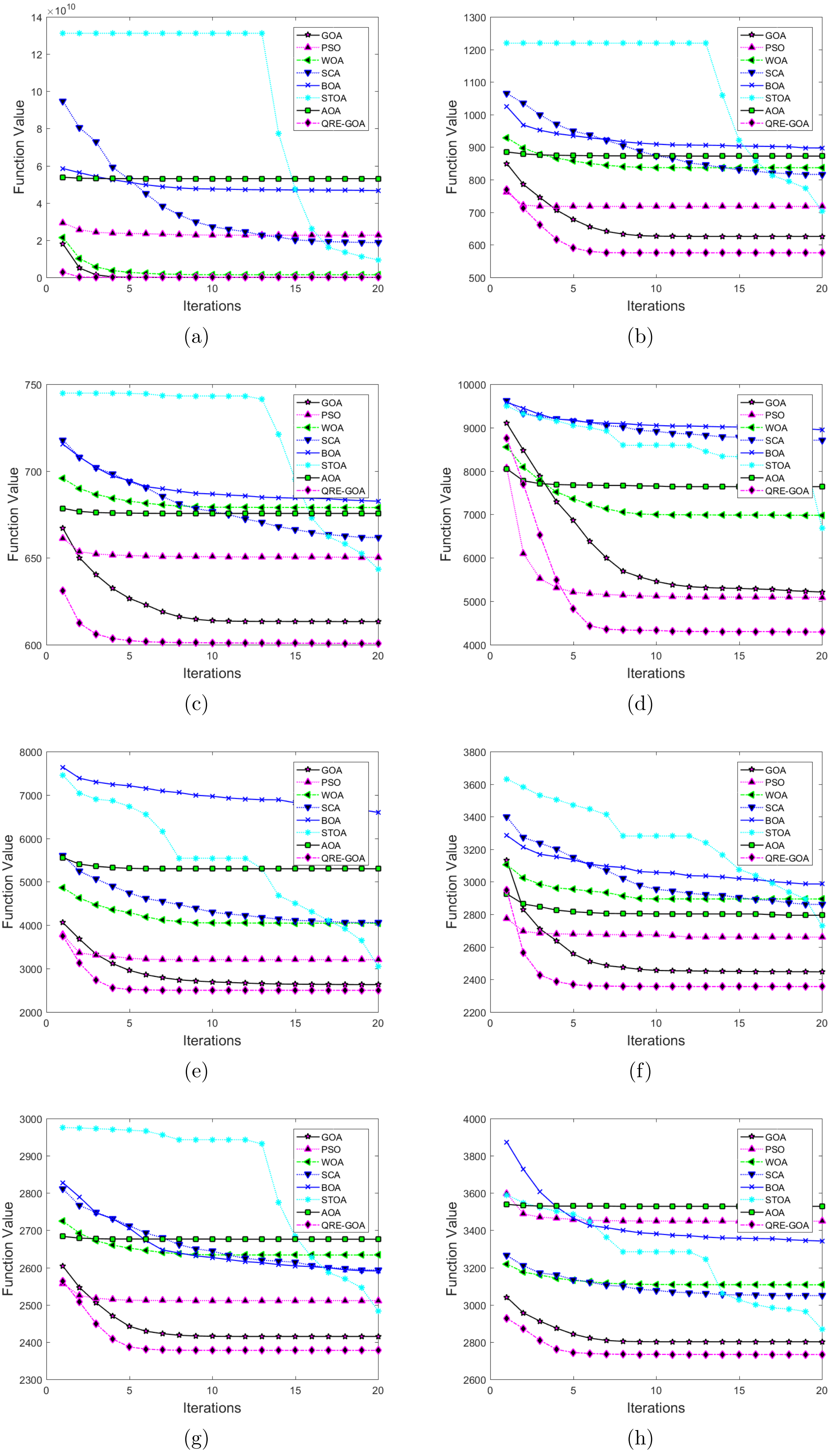
4.4. **Analysis of Algorithm Convergence.** In order to further prove the superiority of the optimized new algorithm QRE-GOA compared with the original algorithm GOA and other algorithms, we compared the convergence of the QRE-GOA with other 7 meta-heuristic algorithms, including the old the algorithm PSO and other emerging algorithms BOA, SCA, WOA, etc, so it can more comprehensively prove the superiority of the QRE-GOA. We randomly selected 1 unimodal function, 3 multimodal functions, 2 mixed functions, and 2 combined functions. Among them, the randomly selected unimodal function is F1, the randomly selected multimodal functions are F5, F6, and F10, the randomly selected mixed functions are F16 and F20, and the finally randomly selected combined functions are F21 and F23. The randomness and fairness properties of the QRE-GOA are more comprehensively proved. From the following 8 images, it can be seen that the convergence speed of QRE-GOA is not only faster than that of the GOA on unimodal functions, multimodal functions, mixed functions, and combined functions, but also faster than the other 5 algorithms. The convergence speed is fast, which shows that the elite selection strategy and the QUATRE algorithm we added have played a significant role in the convergence of the particle swarm, because the QUATRE algorithm can quickly search for the optimal solution, and the elite selection strategy can retain some of the optimal solution. Therefore, QRE-GOA can achieve rapid convergence. In addition, QRE-GOA can not only achieve rapid convergence, but also realize the trend that the fitness value continues to decline after the iteration ends, which proves that our addition of the QUATRE algorithm and the restart mechanism play a vital role, which can make the particles that gradually fall into the local optimal solution escape through the disturbance of QUATRE and the restart strategy, which greatly reduces the situation of falling into the local optimal solution. Please see Figure 1 below for specific data. The purpose of this research is to find a more excellent algorithm based on the GOA, which can improve its ability to find the optimal solution. We discussed in depth the operating mechanism of the QUATRE algorithm, the restart strategy and the elite selection strategy, and we clearly know why the optimized QRE-GOA is better than the original algorithm. And it is hoped that the improved algorithm can be accurately applied to the 3D coverage of the wireless sensor network. The above experiments have proved the superiority of the QRE-GOA from multiple angles and levels [41].

4.5. **Analysis of Significant Differences on 29 Test Functions.** In order to statistically verify the superior results obtained based on the QRE-GOA. By performing the friedman test and the wilcoxon signed-rank test [42], we compared it with the original algorithm GOA and other 4 algorithms, so we can more fully evaluate the performance differences between QRE-GOA and other algorithms, and determine whether these differences are in the statistically significant. Table 4 and Table 5 provide objective data support for these statistical tests. Among them, we still set the number of particle populations to 30, the number of iterations to 1000, and the dimension to 30 dimensions. At the same time, the data with p value greater than 0.05 are bolded, because their results were not statistically significant. The data in Table 4 and Table 5 further help us confirm that the excellent performance of the QRE-GOA algorithm is significant and credible in practical problems. It is obvious that the significance level of QRE-GOA is far better than the original algorithm GOA and other algorithms.

5. **3D Coverage.** In the context of the three-dimensional wireless sensor network coverage problem, the specification of two coordinates and a radius R defines a spherical space. The sensor detection area is represented by the surface of this spherical space. The integration of the QRE-GOA algorithm into the optimization of these coordinates offers a promising avenue to enhance the coverage capabilities of wireless sensor networks. Each deployment strategy is metaphorically treated as a "gannet," and its configuration is articulated through Equation (14). This method of representation facilitates a clearer understanding and description of sensor deployment strategies. The utilization of the QRE-GOA algorithm is anticipated to refine these deployment strategies, ultimately leading to an improvement in the overall coverage performance.

$$[G\mathrm{an}_1^1, G\mathrm{an}_1^2, G\mathrm{an}_2^1, G\mathrm{an}_2^2, \cdots, G\mathrm{an}_i^1, G\mathrm{an}_i^2, \cdots, G\mathrm{an}_N^1, G\mathrm{an}_N^2] \tag{14}$$

i represents the index of the sensor node, and N signifies the total count of sensor nodes. The term $G\mathrm{an}_i^1$ corresponds to the first-dimensional value of the $i-th$ node, while the



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

FIGURE 1. Convergence curves of F1, F5, F6, F10, F16, F20, F21 and F23

TABLE 4. Friedman's test for other comparative algorithms.

Function	Sum of squares	Degree of freedom	Mean squares	p-value
F1	1392.4	5	278.48	1.67308×10^{-21}
F3	1268.8	5	253.76	1.69318×10^{-19}
F4	1326.5	5	265.30	1.96471×10^{-20}
F5	1126.1	5	225.22	3.43763×10^{-17}
F6	1147.5	5	229.50	1.55163×10^{-17}
F7	1238.7	5	247.74	5.20190×10^{-19}
F8	1073.9	5	214.78	2.38776×10^{-16}
F9	1175.4	5	235.08	5.49616×10^{-18}
F10	1209.7	5	241.94	1.53271×10^{-13}
F11	1231.4	5	246.28	6.82855×10^{-19}
F12	1360.9	5	272.18	5.43317×10^{-21}
F13	1358.8	5	271.76	5.87684×10^{-21}
F14	1039.9	5	207.98	8.42298×10^{-16}
F15	1152.7	5	230.54	1.27882×10^{-17}
F16	1075.8	5	215.16	2.22525×10^{-16}
F17	1073.3	5	214.66	2.44151×10^{-16}
F18	1039.1	5	207.82	8.67640×10^{-16}
F19	1135.7	5	227.14	2.40607×10^{-17}
F20	996.7	5	199.34	4.16983×10^{-15}
F21	1111.9	5	222.38	5.82604×10^{-17}
F22	938.0	5	187.60	3.64862×10^{-14}
F23	1292.7	5	258.54	6.94062×10^{-20}
F24	1353.1	5	270.62	7.27223×10^{-21}
F25	1340.8	5	268.16	1.15156×10^{-20}
F26	1296.3	5	259.26	6.06793×10^{-20}
F27	1298.3	5	259.66	5.63142×10^{-20}
F28	1326.4	5	265.28	1.97207×10^{-20}
F29	1204.6	5	240.92	1.85333×10^{-18}
F30	1306.0	5	261.20	4.22450×10^{-20}

second-dimensional value of the $i-th$ node is also referred to as the "$G\mathrm{an}_i^2$". The coverage rate for the $i-th$ round can be computed using Equation (15).

$$rat(k) = \frac{1}{H}\sum_{e=1}^{H}(\sum_{f=1}^{N} F(P_f, C_e)) \tag{15}$$

Here, H denotes the number of pixels representing target objects in the 3D terrain. N refers to the total number of sensor nodes. The function $F(P_f, C_e)$ signifies whether the pixel e is covered by node f. If the Euclidean distance between node P and target object C is less than the radius R, then $F(P_f, C_e)$ is set to 1, otherwise, $F(P_f, C_e)$ is set to 0.

5.1. **Parameter settings.** To verify the efficacy of the QRE-GOA in 3D space, the study used the algorithm to solve the 3D coverage problem, and demonstrates its advantages

Table 5. Wilcoxon signed rank test compared with other algorithms.

Function	GOA	PSO	WOA	AOA	SCA
F1	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F3	**0.0720**	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F4	**0.1264**	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F5	3.9388×10^{-07}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F6	7.8980×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F7	2.5960×10^{-05}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F8	2.7451×10^{-04}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F9	3.4156×10^{-07}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F10	2.2220×10^{-04}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F11	1.4400×10^{-02}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F12	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F13	00.6600×10^{-02}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F14	5.2269×10^{-07}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F15	**0.6359**	6.7956×10^{-08}	9.1728×10^{-08}	3.4995×10^{-06}	6.7956×10^{-08}
F16	0.8400×10^{-02}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F17	**0.0679**	1.6571×10^{-07}	9.1728×10^{-08}	6.7956×10^{-08}	7.8980×10^{-08}
F18	0.3300×10^{-02}	6.7956×10^{-08}	7.8980×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F19	1.2300×10^{-02}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F20	**0.2616**	1.0646×10^{-07}	7.8980×10^{-08}	1.5757×10^{-06}	2.5629×10^{-07}
F21	4.6007×10^{-04}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F22	3.6000×10^{-02}	2.9598×10^{-07}	5.2550×10^{-05}	6.7956×10^{-08}	2.9598×10^{-07}
F23	1.4438×10^{-04}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F24	2.5629×10^{-07}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F25	2.0700×10^{-02}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F26	1.2505×10^{-05}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F27	2.9400×10^{-02}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F28	3.1500×10^{-02}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F29	**0.0565**	6.7956×10^{-08}	7.8980×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}
F30	1.6098×10^{-04}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}	6.7956×10^{-08}

in this field through different parameter settings. Specifically, in this study, we apply the QRE-GOA to solve the coverage problem in a three-dimensional area, that is, how to efficiently deploy wireless sensors to cover the entire area. To ensure a fair comparison, we set the number of particles of PSO, APSO, FMO and QRE-GOA to 100, and made multiple settings for the number of nodes (N), which were 30, 35, 40, and 45, respectively. In addition, we also adjusted the radius of the sensor and tried different values, including 3, 7 and 10. The comparison results are shown in Table 6. In these experiments, the dimension (D) of the number of nodes is twice the number of nodes, $D = N * 2$. In order to simulate the real situation, we randomly arranged wireless sensors [43] in a 50×50 mountain area. Through these experiments, we aim to demonstrate the effectiveness and advantages of the QRE-GOA in solving 3D coverage problems [44, 45]. The simulation results are shown in Figure 2.

5.2. **Results.** In order to ensure the accuracy of the experimental results, each algorithm was run independently 10 times, and its average value was calculated in Table 6. The results for different number of nodes (N) and coverage radius (R) are listed in the table. Among them, N represents the number of nodes, and R represents the coverage radius.

Table 6. Coverage of different algorithms.

N	R	PSO(%)	APSO(%)	FMO(%)	QRE-GOA(%)
30	3	14.46	13.61	14.99	**15.51**
	5	46.61	40.97	46.51	**48.66**
	7	74.51	63.62	74.27	**77.33**
	10	94.82	85.23	94.86	**96.87**
35	3	17.32	15.60	17.25	**18.02**
	5	51.93	45.72	51.47	**53.67**
	7	79.18	67.60	79.16	**82.14**
	10	96.61	87.57	96.69	**98.32**
40	3	19.34	17.30	19.35	**20.23**
	5	56.35	49.23	56.32	**58.25**
	7	82.94	73.53	83.13	**86.04**
	10	97.96	90.00	98.01	**98.96**
45	3	21.44	19.18	21.44	**22.39**
	5	60.16	53.24	60.30	**62.37**
	7	86.24	76.89	86.32	**88.54**
	10	98.68	92.22	98.68	**99.38**

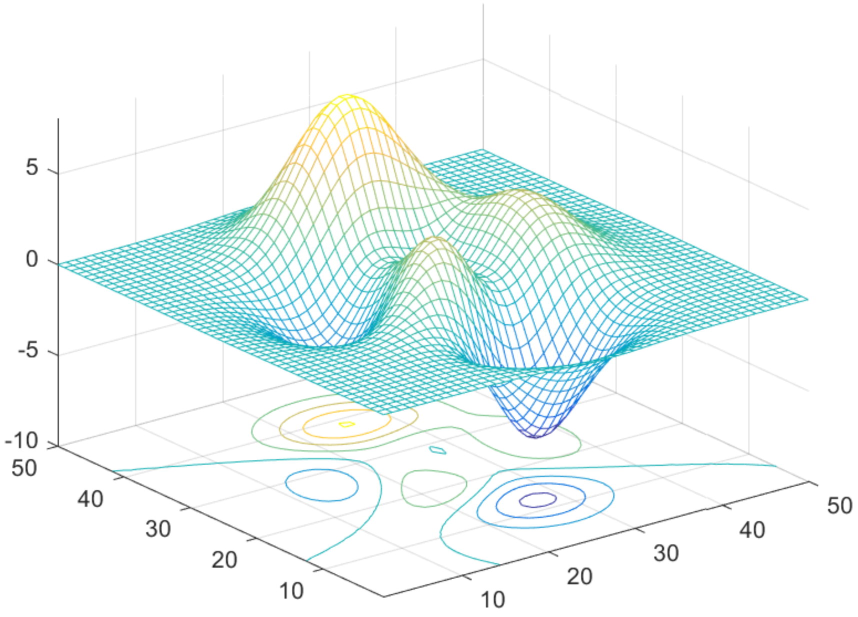


Figure 2. The simulated 3D terrain

By comparing the experimental results, we can see that the QRE-GOA has obvious advantages in 3D coverage. These results will help to verify the effectiveness of the QRE-GOA in practical applications, and provide a new perspective for the solution of 3D coverage problems.

6. **Conclusions.** In this paper, we present a series of improvements to GOA aimed at improving its optimization performance. First, we introduce new elements, including the QUATRE algorithm, restart strategy, and elite selection. These improvements dramatically change the position change strategy of the particles and allow the algorithm to explore the solution space more flexibly, thus further improving its optimization capability. In addition, to study the optimization performance of the new algorithm, we also conducted a comparative analysis with other well-known algorithms in 29 different types of test functions, thus affirming the strong competitiveness of the new algorithm. Finally, we apply QRE-GOA to solve the 3D coverage problem and demonstrate through simulation that QRE-GOA achieves significantly higher coverage compared with other algorithms.

REFERENCES

[1] J. Derrac, S. García, D. Molina, and F. Herrera , "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[2] A. Slowik, and H. Kwasnicka , "Nature inspired methods and their industry applications—Swarm intelligence algorithms," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1004–1015, 2017.

[3] J. Tang, G. Liu, and Q. Pan, "A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 10, pp. 1627–1643, 2021.

[4] Y. Tan, and K. Ding, "A survey on GPU-based implementation of swarm intelligence algorithms," *IEEE Transactions on Cybernetics*, vol. 46, no. 9, pp. 2028–2041, 2015.

[5] G.-G. Wang, and Y. Tan, "Improving metaheuristic algorithms with information feedback models," *IEEE Transactions on Cybernetics*, vol. 49, no. 2, pp. 542–555, 2017.

[6] Y. Tang, J. Chen, and J. Wei, "A surrogate-based particle swarm optimization algorithm for solving optimization problems with expensive black box functions," *Engineering Optimization*, vol. 45, no. 5, pp. 557–576, 2013.

[7] C.-M. Chen, S. Lv, J. Ning, and J. M.-T. Wu, "A Genetic Algorithm for the Waitable Time-Varying Multi-Depot Green Vehicle Routing Problem," *Symmetry*, vol. 15, no. 1, 124, 2023.

[8] X.-D. Li, J.-S. Wang, W.-K. Hao, M. Zhang, and M. Wang, "Chaotic arithmetic optimization algorithm," *Applied Intelligence*, vol. 52, no. 14, pp. 16718–16757, 2022.

[9] L.-L. Kang, R.-S. Chen, N.-X. X, Y.-C. Chen, Y.-X. Hu, and C.-M. Chen, "Selecting Hyper-Parameters of Gaussian Process Regression Based on Non-Inertial Particle Swarm Optimization in Internet of Things," *IEEE Access*, vol. 7, pp. 59504-59513, 2019.

[10] L. Abualigah, and A. Diabat, "Advances in sine cosine algorithm: a comprehensive survey," *Artificial Intelligence Review*, vol. 54, no. 4, pp. 2567–2608, 2021.

[11] J.-S. Pan, L.-G. Zhang, R.-B. Wang, V. Snášel, and S.-C. Chu, "Gannet optimization algorithm: A new metaheuristic algorithm for solving engineering optimization problems," *Mathematics and Computers in Simulation*, vol. 202, pp. 343–373, 2022.

[12] J.-S. Pan, B. Sun, S.-C. Chu, M. Zhu, and C.-S. Shieh, "A parallel compact gannet optimization algorithm for solving engineering optimization problems," *Mathematics*, vol. 11, no. 2, 439, 2023.

[13] S. Das, and P.-N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2010.

[14] T.-Y. Wu, H.-Nan. Li, and S.-C. Chu, "CPPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps," *Mathematics*, vol. 11, no. 9, 1977, 2023.

[15] F. Kolahan, and M. Liang, "Optimization of hole-making operations: a tabu-search approach," *International Journal of Machine Tools and Manufacture*, vol. 40, no. 12, pp. 1735–1753, 2000.

[16] F.-Q. Z, T.-Y. Wu, Y. Wang, R. Xiong, G.-Y. Ding, P. Mei, and L.-Y. L, "Application of quantum genetic optimization of LVQ neural network in smart city traffic network prediction," *IEEE Access*, vol. 8, pp. 104555-104564, 2020.

[17] N. Mansouri, and A.-M. Sharafaddini, "An efficient gannet optimization algorithm for feature selection based on sensitivity and specificity," *Journal of Algorithms and Computation*, vol. 54, no. 2, pp. 49–69, 2022.

[18] D.-R Rao, T.-J Prasad, and M.-N Prasad, "Gannet optimization algorithm enabled framework for spectrum sensing in OFDM based CR network," *Wireless Networks*, pp. 1–10, 2023.

[19] S. Zhao, T. Zhang, S. Ma, and M. Wang, "Sea-horse optimizer: a novel nature-inspired meta-heuristic for global optimization problems," *Applied Intelligence*, vol. 53, no. 10, pp. 11833–11860, 2023.

[20] R.-B. Wang, R.-B. Hu, F.-D. Geng, and L. Xu, "Optimizing the Layout of Nucleic Acid Test Sites for COVID-19 Based on Gannet Optimization Algorithm," *Advances in Smart Vehicular Technology, Transportation, Communication and Applications: Proceedings of VTCA 2022*, pp. 453–462, 2023.

[21] J.-S. Pan, F.-F. Liu, J. Wu, T.-S. Pan and S.-C. Chu, "Research on Gannet Optimization Algorithm and Its Application in Traveling Salesman Problem," *Advances in Smart Vehicular Technology, Transportation, Communication and Applications: Proceedings of VTCA 2022*, pp. 343–352, 2023.

[22] J.-B. Su, R.-B. Wang, F.-D. Geng, Q. Wei, and L. Xu, "A Parallel Gannet Optimization Algorithm with Communication Strategies (PGOA)," *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 71–80, 2022.

[23] X.-S. Yang, "Review of meta-heuristics and generalised evolutionary walk algorithm," *International Journal of Bio-Inspired Computation*, vol. 3, no. 2, pp. 77–84, 2011.

[24] J.-S. Pan, R.-Y. Wang, S.-C. Chu, K.-K. Tseng, and F. Fan, "A Quasi-Affine Transformation Evolutionary Algorithm Enhanced by Hybrid Taguchi Strategy and Its Application in Fault Detection of Wireless Sensor Network," *Symmetry*, vol. 15, no. 4, 795, 2023.

[25] Z. Meng, J.-S. Pan, and H. Xu, "QUasi-Affine TRansformation Evolutionary (QUATRE) algorithm: A cooperative swarm based algorithm for global optimization," *Knowledge-Based Systems*, vol. 109, pp. 104–121, 2016.

[26] Z.-G. Du, J.-S. Pan, S.-C. Chu, H.-J. Luo and P. Hu, "Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks," *IEEE Access*, vol. 8, pp. 8583–8594, 2020.

[27] M. Arunkumar, and K.-A. Kumar, "GOSVM: Gannet optimization based support vector machine for malicious attack detection in cloud environment," *International Journal of Information Technology*, vol. 15, no. 3, pp. 1653–1660, 2023.

[28] T. Prabhakar, T.-V.-M. Rao, B. Maram, and D. Chigurukota, "Exponential gannet firefly optimization algorithm enabled deep learning for diabetic retinopathy detection," *Biomedical Signal Processing and Control*, vol. 87, 105376, 2024.

[29] S. Krishnamoorthy, Y. Weifeng, J. Luo, and S. Kadry, "GO-DBN: Gannet Optimized Deep Belief Network Based wavelet kernel ELM for Detection of Diabetic Retinopathy," *Expert Systems with Applications*, vol. 229, 120408, 2023.

[30] K. Bhamidipati, S. Muppidi, P.-V Reddy, "Soil Moisture and Heat Level Prediction for Plant Health Monitoring Using Deep Learning with Gannet Namib Beetle Optimization in IoT," *Applied Biochemistry and Biotechnology*, pp. 1–29, 2023.

[31] C.-D. Tarantilis, and C.-T. Kiranoudis, "A meta-heuristic algorithm for the efficient distribution of perishable foods," *Journal of food Engineering*, vol. 50, no. 1, pp. 1–9, 2001.

[32] T.-Y. Wu, A.-K. Shao, and J.-S. Pan, " CTOA: Toward a Chaotic-Based Tumbleweed Optimization Algorithm, " *Mathematics*, vol. 11, no. 10, 2339, 2023.

[33] B. Zhao, T.-W. Sung, and X. Zhang, "A quasi-affine transformation artificial bee colony algorithm for global optimization," *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 3, pp. 5527–5544, 2021.

[34] Y. Shi, H. Liu, L. Gao, and G. Zhang, "Cellular particle swarm optimization," *Information Sciences*, vol. 181, no. 20, pp. 4460–4493, 2011.

[35] D. Karaboga, and B. Akay, "A comparative study of artificial bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.

[36] J. Nayak, H. Swapnarekha, B. Naik, G. Dhiman, and S. Vimal, "25 years of particle swarm optimization: Flourishing voyage of two decades," *Archives of Computational Methods in Engineering*, vol. 30, no. 3, pp. 1663–1725, 2023.

[37] S. Chakraborty, A.-K. Saha, S. Sharma, R. Chakraborty and S. Debnath, "A hybrid whale optimization algorithm for global optimization," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 1, pp. 431–467, 2023.

[38] M. Li, G. Xu, L. Zeng, and Q. Lai, "Hybrid whale optimization algorithm based on symbiosis strategy for global optimization," *Applied Intelligence*, vol. 53, no. 13, pp. 16663–16705, 2023.

[39] L. Abualigah, and A. Diabat, "Advances in sine cosine algorithm: a comprehensive survey," *Artificial Intelligence Review*, vol. 54, no. 4, pp. 2567–2608, 2021.

[40] P. Chakraborty, S. Nama, and A.-K. Saha, "A hybrid slime mould algorithm for global optimization," *Multimedia Tools and Applications*, vol. 82, no. 15, pp. 22441–22467, 2023.

[41] D.-K. Sah, S. Srivastava, R. Kumar, and T. Amgoth, "An energy efficient coverage aware algorithm in energy harvesting wireless sensor networks," *Wireless Networks*, vol. 29, no. 3, pp. 1175–1195, 2023.

[42] M. Kitani, and H. Murakami, "One-sample location test based on the sign and Wilcoxon signed-rank tests," *Journal of Statistical Computation and Simulation*, vol. 92, no. 3, pp. 610–622, 2022.

[43] Q.-W. Chai, S.-C. Chu, J.-S. Pan, and W.-M. Zheng, "Applying Adaptive and Self Assessment Fish Migration Optimization on Localization of Wireless Sensor Network on 3-D Terrain," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 11, no. 2, pp. 90–102, 2020.

[44] A. Katti, "Target coverage in random wireless sensor networks using cover sets," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 3, pp. 734–746, 2022.

[45] R.-B. Wang, W.-F. Wang, L. Xu, J.-S. Pan, and S.-C. Chu, "Improved DV-Hop based on parallel and compact whale optimization algorithm for localization in wireless sensor networks," *Wireless Networks*, vol. 28, no. 8, pp. 3411–3428, 2022.