Computing and Informatics, Vol. 42, 2023, 1305–1338, doi: 10.31577/cai_2023_6_1305

QUANTUM-BEHAVED BAT ALGORITHM COMBINING CONVERGENCE FACTOR AND SELF-LEARNING MUTATION STRATEGIES FOR OPTIMIZATION

Yao Chen*

Xi'an Key Laboratory of Human-Machine Integration and Control Technology for Intelligent Rehabilitation
School of Computer Science, Xijing University
Shaanxi 710123, P.R. China
e-mail: `chenyxjxy@126.com`

Si Chen

Department of Applied Mathematics, School of Science
Xi'an Shiyou University
Shaanxi 710065, P.R. China
e-mail: `chen1991si@126.com`

Jian-Hui Li

Xi'an Key Laboratory of Human-Machine Integration and Control Technology for Intelligent Rehabilitation
School of Computer Science, Xijing University
Shaanxi 710123, P.R. China
e-mail: `15829627364@163.com`

Abstract. Quantum-behaved Bat Algorithm (QBA) has been successfully applied as an optimal technique for dealing with a variety of optimization problems. Nevertheless, QBA suffers from similar problems as other swarm intelligent algorithms,

* Corresponding author

such as poor exploration search and falling into local optima in certain conditions. Aiming at these shortcomings, an improved algorithm that combines convergence factor and gold sinusoidal self-learning mutation strategies (CGQBA) is proposed. A directional convergence factor is designed for the global position update process, it can improve the exploration search ability of the algorithm. Meanwhile, a self-learning predictive mutation mechanism is added to the algorithm. It contributes to the algorithm to jump out of the local extremum. The improved CGQBA algorithm is tested on 20 test functions with different characteristics in the numerical simulation experiments. The results and statistical tests show that CGQBA algorithm has better convergence speed, accuracy and stability. What is more, the multi-threshold image segmentation is modelled as an optimization problem, CGQBA algorithm is applied to the optimization problem to further verify the effectiveness and practicability in the real-world optimization. The results compared with three classical segmentation methods illustrate that CGQBA algorithm can effectively solve the image segmentation problem. It has a better segmentation effect and anti-noise ability.

Keywords: Swarm intelligence, quantum-behaved bat algorithm, convergence analysis, optimization

1 INTRODUCTION

With the development of computational intelligence, since Eberhart and Kennedy proposed the particle swarm algorithm [1] (PSO) based on bird flock behavior, there is a wave of invention for swarm intelligence algorithms. Scholars have proposed many intelligent optimization algorithms to solve complex optimization problems. The ant colony algorithm was proposed by Dorigo et al. to simulate the foraging behavior of ants [2] (ACO). Passino proposed a bacterial foraging algorithm [3] (BFO) by imitating the behavior of Escherichia coli to devour food in the human gut. In order to solve the multivariate function optimization problem, Karaboga and Basturk published the artificial bee colony algorithm [4] (ABC). British scholar Yang proposed the bat algorithm (BA) [5] and the cuckoo search algorithm [6] (CS) in 2010 and 2013, respectively. In recent years, scholars have successively proposed the gray wolf optimization algorithm [7] (GWO), the whale optimization algorithm [8] (WOA), and the sparrow search algorithm [9] (SSA). The above-mentioned swarm intelligent algorithms generally have the common characteristics of randomness, parallelism and distribution in the optimization process, and they have the advantages of wide application range and good performance for optimization problems.

As an important member of the swarm intelligent algorithm, BA has been widely and successfully applied to various complex optimization problems. For example, Long and Zhang applied the BA algorithm to find the solution of constrained optimization problems [10]. Dai and Luo applied the BA algorithm into wireless network traffic prediction [11]. Li et al. applied the BA algorithm to the path optimization of intelligent robots [12]. Rauf et al. proposed an enhanced BA algorithm and applied it to predict the number of COVID-19 cases [13]. In a large number of application practices, researchers have proposed many improved versions of the BA algorithm to balance the global exploration and local search capabilities of the algorithm. The corresponding improvement directions of the BA algorithm can be divided into the following categories: The first category is integrating other algorithms to obtain the advantages and make up for the shortcomings, such as combining the BA algorithm with the differential evolution algorithm and particle filter algorithm [14, 15]. Second one is adding new learning mechanisms and optimizing search methods, such as the improved BA algorithm [16] based on the hybrid optimization strategy, introducing the Sobol sequence and the intermittent Lévy jump strategy [17] into the process of speed and position update. What is more, there are other aspects to improve the performance of the BA algorithm. For example, Li et al. extended the BA algorithm to the quantum space and proposed a BA algorithm with quantum behavior [18] (quantum-behaved bat algorithm, QBA). In the past two years, some scholars have proposed improved quantized bat algorithm. For example, Zhu et al. proposed a QBA algorithm with an average best position orientation [19]. Rugema et al. proposed a novel Cauchy-Gaussian quantum behavioral bat algorithm and applied it to solve the problem of economic load distribution [20]. Yang et al. applied the QBA algorithm to solve the multi-stacker collaborative scheduling optimization problem [21]. Gao et al. applied the QBA algorithm to the cognition of infinite energy harvesting to obtain the optimal solution for cooperative transmission of energy and information [22].

Despite their efforts, the improvements were limited and there is still room for improvement. Therefore, this paper proposes an improved QBA with a directional convergence factor and self-learning mutation strategy, and such strategies were not used in the original BA. First of all, the directional convergence factor is designed in the process of updating the global position of bat individuals. It can dynamically adjust the search step size of individual bats, and it can enhance the diversity of bats' location information. Secondly, a gold sinusoidal self-learning mutation strategy is designed, which objectively determines whether the algorithm needs to be mutated by the fitness value of the objective function. If the solution falls into local extrema, or the algorithm does not search in a better direction, it will adaptively trigger the mutation strategy and integrate the golden sine algorithm to generate mutation.

The paper is thus organized as follows. The relevant content about quantum-behaved BA is introduced in Section 2, followed by a detailed presentation of improvements of the CGQBA in Section 3. Section 4 gives a global convergence analysis of the update algorithm. Section 5 presents numerical experiments on 20 classic benchmarks and a real-world classic image multi-threshold segmentation application was shown in Section 6. Finally, we outline some conclusions with brief discussions in Section 7.

2 QUANTUM-BEHAVED BAT ALGORITHM (QBA)

The BA algorithm was proposed by British scholar Xin-she Yang, and was inspired by the predation of bat groups through echolocation systems. Li et al. proposed a new version of BA algorithm (QBA) with quantum behavior according to the relevant theory of quantum systems. The search behavior of BA algorithm is improved to the quantum space, and the aggregation limitation in the search process of BA algorithm is solved by using the state superposition principle of the quantum system.

The principle of QBA algorithm is to regard each bat in the population as a particle in the quantum space. By using the potential field effect of the particle, the algorithm has an aggregated state and can appear in any area of the entire feasible search space with a certain probability. In short, compared with the BA algorithm, the QBA algorithm deletes the velocity update formula and resets the position update formula of the algorithm according to the theory of quantum systems. The search steps for solving the objective function are consistent with the BA algorithm. The steps of the QBA algorithm can be described as follows:

Step 1: Initialize the basic parameters and randomly generate the position information x_i of the bats. The basic parameters include: bat population size N, pulse frequency r_i, loudness A_i, maximum number of iterations T_{max}, frequency f_i and frequency range $[f_{min}, f_{max}]$.

Step 2: Calculate the fitness value of the objective function on the initialized position information, and then find the global optimal position P_g.

Step 3: Update the pulse frequency f_i and the position information x_i according to the iterative formula of the QBA algorithm. The specific iterative formula is as follows:

$$x_i^{t+1} = \left(P \pm \alpha|P - x_i^t| \cdot \ln\left(\frac{1}{u}\right)\right) \cdot f_i^t, \tag{1}$$

$$f_i = f_{min} + (f_{max} - f_{min}) \cdot \beta_*, \tag{2}$$

$$P = \phi_1 P_i + (1 - \phi_1) P_g, \tag{3}$$

$$u = rand(0, 1), \tag{4}$$

where $\beta_* \in [0, 1]$, ϕ_1 obeys a random distribution on range of $(0, 1)$, P_i represents the current optimal position, and P_g represents the global optimal position.

Step 4: Generate a random number $rand_1$, if the random number $rand_1 > r_i^t$ (the formula of r_i^t is shown as Equation (7)), then generate a new solution around the neighborhood of the optimal solution x_* through random disturbance:

$$X_{new} = p_{best} + \varepsilon A^t, \tag{5}$$

where $rand_1 \in [0,1]$ obeys a uniform distribution, $\varepsilon \in [-1,1]$ is a random vector, and A^t is the average loudness of bats.

Step 5: Generate a random number $rand_2$. If the random number $rand_2 < A_i$ and $f(X_{new}) < f(x_*)$, then accept the new solution. At this time, the bat is improved, and the pulse frequency r_i and loudness A_i are updated according to the following formulas:

$$A_i^{t+1} = \alpha A_i^t, \tag{6}$$

$$r_i^{t+1} = r_i^0 \left[1 - \exp(-\gamma t)\right], \tag{7}$$

where $0 < \alpha < 1$, $\gamma > 0$, both of them are constants.

Step 6: Update the optimal solution and the global optimal solution according to the fitness value of the objective function.

Step 7: Repeat Steps 3-6 until the global optimum is output when the end conditions are met, and the optimization ends.

3 IMPROVED QBA ALGORITHM (CGQBA)

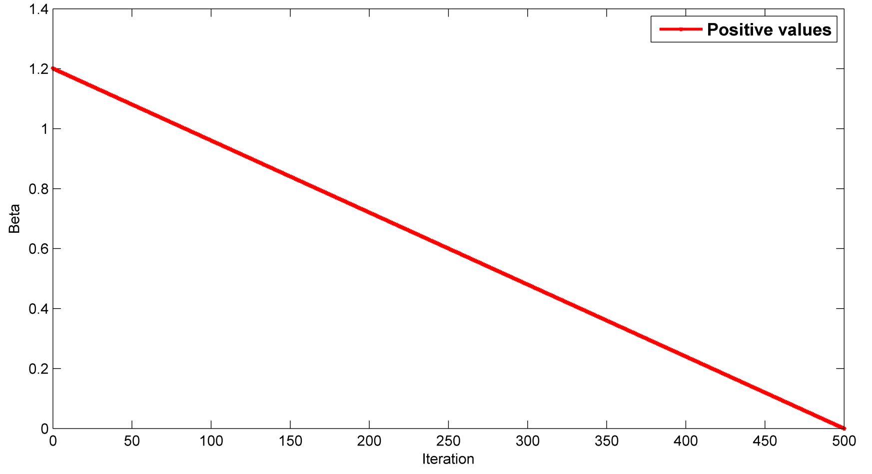
3.1 Introduced Convergence Factor with Direction

In order to improve the global exploration ability of the QBA algorithm, a shrinkage factor is designed as a new strategy, and this directional convergence factor β is added to the global position update process. In the early stages of the update process, a larger value of $|\beta|$ enables the bat to obtain a larger step size, which is conducive to a large-scale global search and has a stronger global exploration ability. As $|\beta|$ decreases, its value begins to gradually shrink to a smaller value. At this time, the step size becomes smaller, which tends to refine the local search, and it is beneficial to improve the convergence accuracy in the later stage of the algorithm. The mathematical expression for β is shown as follows:

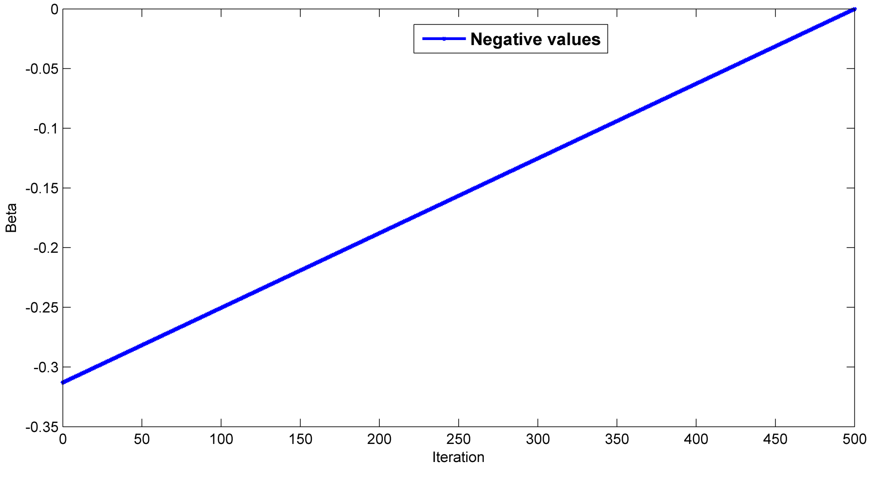
$$\begin{cases} \beta(t) = 2\delta\left(1 - \frac{t}{T_{max}}\right), \\ \delta = 2 * rand - 1, \end{cases} \tag{8}$$

where t represents the current number of iteration, T_{max} represents the maximum number of iteration, δ is the direction deviation coefficient, and $\delta \in (-1,1)$. $rand \in [0,1]$ obeys a uniform distribution. The convergence factor β gradually approaches 0 with the increase of the number of iterations. β has two random directions, positive and negative, and its values are shown in Figure 1.

By analyzing the mathematical expression and the value figure of β, we can see that the value of β is in a random positive and negative state. In a certain search process, β can not only dynamically adjust the search step size of individual bats, but also increase the diversity of individual bat location information. After



a) Positive values of β



b) Negative values of β

Figure 1. The values of convergence factor β

introducing the shrinkage factor, the global position update formula of the algorithm is expressed as:

$$X_i^{t+1} = \beta \cdot \left[\left(P \pm \alpha |P - X_i^t| \cdot \ln \left(\frac{1}{u} \right) \right) \cdot f_i^t \right]. \tag{9}$$

3.2 Gold Sinusoidal Self-Learning Mutation Strategies

Just like other swarm intelligent algorithms, the QBA algorithm has randomness in the search process, which leads to the disadvantage that it may easily fall into local extreme values in some cases. In response to this shortcomming, this paper proposed a self-learning predictive mutation mechanism to the QBA algorithm to strengthen the algorithm's ability to jump out of local extrema. The proposed adaptive mutation mechanism mainly includes two parts: self-learning prediction operation and golden sine mutation mechanism.

3.2.1 Self-Learning Predictive Operation

In order to enhance the utilization of historical optimization information and the self-learning ability of bats in the search process, the global optimal position of the bat population is used as the criterion for judging whether the current optimization result needs to be mutated. In the optimization process, the optimal solution searched by the i^{th} bat in the t^{th} iteration is recorded as p_i^t. The mathematical model of the adaptive pre-judgment operation can be expressed as two modules: the global optimal evaluation module of the bat population and the pre-judgment conditions.

First of all, the global optimal position found by n bat individuals under the current number of iterations is evaluated in the global optimal evaluation module according to the fitness value of the objective function, and its expression is:

$$f_{gbest}^t = \min_{i=1}^{n} f_{p_i}^t, \tag{10}$$

where $f_{p_i}^t$ represents the fitness value of the objective function corresponding to the optimal solution searched by the i^{th} bat individual in the t^{th} iteration. This evaluation module records the objective function fitness value f_{gbest}^t corresponding to the optimal solution searched after t iterations of the entire population, and uses it as the standard for pre-judgment operations.

The second module is the pre-judgment condition. In this module, the currently searched solution is compared with the recorded fitness value of the global optimal solution to form a pre-judgment condition, and its expression is:

$$f_{t+1} \geq f_{gbest}^t \tag{11}$$

where f_{t+1} represents the objective function fitness value of the solution searched in the $(t+1)^{\text{th}}$ iteration.

The above formula represents that the fitness value of the objective function corresponding to the optimal solution searched by the $(t+1)^{\text{th}}$ iteration is inferior to the global optimal solution searched after the t^{th} iteration, which means that the optimization performance of the current bat population is not good. The search process may get stuck in a state of local extrema. If this condition is satisfied, the algorithm needs to perform mutation perturbation on the current solution. On the contrary, that is, $f_{t+1} < f_{gbest}^t$, which means that the current optimization process is developing for the better, and no mutation disturbance is required, and the next iterative search can be continued.

3.2.2 Golden Sine Variation Mechanism

After the above pre-judgment operation, if the mutation conditions are met, the mutation mechanism will be adaptively triggered. The mutation mechanism designed in this section is inspired by the Golden Sine Algorithm (Golden-SA), which was proposed by Tanyildizi and Demir in 2017 [23]. The core update formulas of the algorithm are shown as follows:

$$\begin{cases} x_i^{t+1} = x_i^t * |\sin(R_1)| + R_2 * \sin(R_1) * |\theta_1 * P_i^t - \theta_2 * x_i^t|, \\ \theta_1 = a * (1-\tau) + b * \tau, \\ \theta_2 = a * \tau + b * (1-\tau), \\ \tau = \frac{\sqrt{5}-1}{2}, \end{cases} \tag{12}$$

where $R_1 \in [0, 2\pi]$, $R_2 \in [0, \pi]$, the two are random numbers in the corresponding range. θ_1 and θ_2 represent the proportional coefficient of the golden ratio; τ is the golden ratio.

In the local search of the QBA algorithm, the random perturbation in the vicinity of the current solution to generate a new solution is changed to use the Golden-SA to generate a new solution. Such an improved mechanism can combine the Golden-SA algorithm into the QBA algorithm as an optimized local operator, so that the two algorithms can be organically integrated, and the search advantage of the Golden-SA algorithm can be used to make up for the lack of the QBA algorithm's reduced convergence accuracy in the later iteration.

Through comprehensive analysis, it is easy to find that the self-learning predictive variation mechanism of the optimal design replaces the part of the original QBA algorithm that determines whether to randomly generate a new solution by comparing the size of random numbers. The generation of the new solution no longer simply depends on the comparison between the random number $rand_2$ and the loudness A_i, but is based on the comparison of the fitness value of the current solution and the global optimal solution to objectively and rationally judge whether the current solution is excellent and whether it is necessary to produce a new solution. In this way, the self-learning mechanism first actually judges and then determines whether mutation is needed, and it can help the algorithm to iterate towards the global optimum. Such an optimized design can improve the performance of the QBA algorithm effectively.

3.3 Operation Steps of CGQBA Algorithm

The QBA algorithm (CGQBA) improved by the above convergence factor and gold sinusoidal self-learning mutation mechanism has the main difference in its operation steps compared with the QBA algorithm: the convergence factor is added to the position update Equation (1) in Step 3, and predictive mutation mechanism is introduced in Step 4. In addition, the original random number size comparison is changed to a pre-judgment condition based on the fitness value of the objective function. If the condition is met, the mutation operation is triggered adaptively, and the Golden-SA is integrated to implement the mutation of the new solution. The rest of the parts are consistent with the QBA algorithm, so the specific operation steps will not be repeated. The flow chart of the CGQBA algorithm is shown in Figure 2.

4 CONVERGENCE ANALYSIS OF CGQBA ALGORITHM

Aiming at the convergence problem of CGQBA algorithm, the method of probability measure is used to prove it. For the convenience of analysis, some related concepts are given first.

4.1 Basic Concepts and Convergence Criteria of Global Search Algorithms

Definition 1. An algorithm is called a global search algorithm if it can guarantee to find the global optimal solution of the desired objective function.

Definition 2. Let the objective function of an optimization problem be f, the solution space of the objective function is from R^n to R, and S is a subset of R^n. There is a point Z in S that can minimize the value of the objective function or generate an acceptable infimum of the objective function on S.

Assumption 1. $f(H(z,\zeta)) \leq f(z)$, if $\zeta \in S$, then $f(H(z,\zeta)) \leq f \leq f(\zeta)$, where H represents a function that generates a solution in the solution space of the objective function, and this function can ensure that the new solution generated is better than the current solution. In order to guarantee the correct operation of the optimization algorithm, H must satisfy Assumption 1. Z represents a minimum value in S, and ζ represents several feasible solutions obtained by the optimization algorithm on S. If a random search algorithm has global convergence, it means that the sequence $f(z_t)_{(t=1)}^{\infty}$ converges to the infimum of the objective function f on S. At the same time, in order to avoid the ill-conditioned situation, the search target is changed to search to the essential infimum Φ, $\Phi = inf(n : q[z \in S | f(z) < n] > 0)$, $q[A]$ represents the Lebesgue measure on the set A.

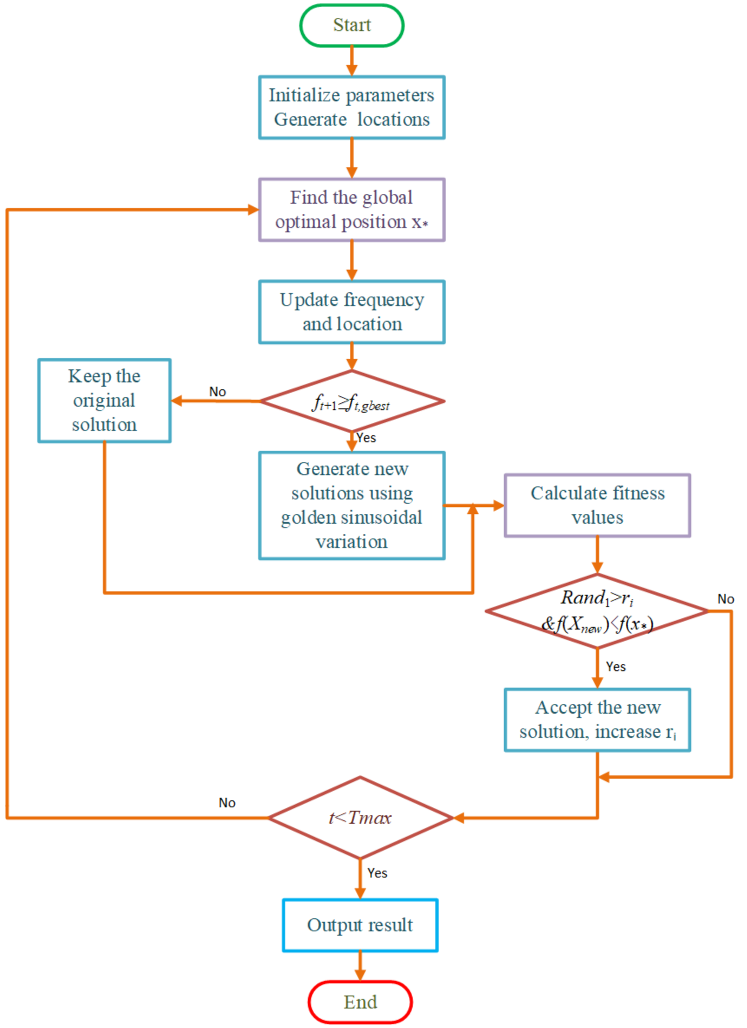


Figure 2. Flowchart of the CGQBA algorithm

Assumption 2. For the set S, A is any Borel subset on it, if the Lebesgue measure $q[A] > 0$ of the subset A, then there are:

$$\prod_{t=0}^{\infty}(1 - \mu_t[A]) = 0, \tag{13}$$

where $\mu_t[A]$ represents the probability of A obtained from measure μ_t.

Theorem 1. (Sufficient and Necessary Conditions for Global Convergence of Random Search Algorithms) [24] The objective function f is a measurable function, S is a measurable subset on R^n, and both Assumption 1 and 2 are satisfied. If the sequence of solutions generated by the search algorithm is set to $\{z^t\}_{t=0}^{\infty}$, then:

$$\lim_{t \to +\infty} B[z_t \in R_\varepsilon] = 1, \tag{14}$$

where $B[z_t \in R_\varepsilon]$ denotes the probability of the solution $z_t \in R_\varepsilon$ generated by the search algorithm at the t^{th} iteration, R_ε denotes the acceptable region of the algorithm, and $R_\varepsilon = z \in S | f(z) < \Phi + \varepsilon, \varepsilon > 0)$.

4.2 Global Convergence of CGQBA Algorithm

Next, by proving that the CGQBA algorithm proposed in this paper can satisfy both Assumption 1 and 2, and then use the necessary and sufficient conditions for the global convergence of the random search algorithm to prove that CGQBA can guarantee the convergence to the global optimal solution.

Lemma 1. CGQBA algorithm can satisfy Assumption 1.

Proof. According to the solution steps of the CGQBA algorithm, it is easy to know that the update method of the current optimal position P_i and the global optimal position P_g of the bat individual with quantum behavior can be expressed as:

$$P_i^{t+1} = \begin{cases} x_i^{t+1}, & f(x_i^{t+1}) < f(P_i^t), \\ P_i^t, & f(x_i^{t+1}) \geq f(P_i^t), \end{cases} \tag{15}$$

$$P_g^{t+1} = \arg \min_{1 \leq i \leq N} f(P_i^t). \tag{16}$$

Thus, the mathematical description of the function H in the entire solution space in the CGQBA algorithm can be written as:

$$H(P_g^t, x_i^t) = \begin{cases} P_g^t, & f(F(x_i^t) \geq f(P_g^t), \\ F(x_i^t), & f(F(x_i^t) < f(P_g^t), \end{cases} \tag{17}$$

where $F(x_i^t)$ represents the specific objective function used in the solution, x_i^{t+1} represents the result obtained after applying the function F, and $x_i^{t+1} = F(x_i^t)$. A series of optimal positions produced by all bat individuals from the first iteration to the t^{th} iteration can be expressed as the sequence $\{P_g^t\}_{t=0}^{t}$. According to the definition of the CGQBA algorithm, this optimal position sequence is obviously monotonic. Therefore, the function H defined in the CGQBA algorithm can satisfy Assumption 1. □

Lemma 2. CGQBA algorithm can satisfy Assumption 2.

Proof. According to the relevant content of quantum theory introduced in the description of the QBA algorithm [18], it can be obtained that the probability density function of the d^{th} dimension of the bat i at the t^{th} iteration of the CGQBA algorithm can be written as:

$$Q(x_{i,d}^t) = \frac{1}{L(i,d,t)} \exp\left(-2|x_{i,d}^t - p_{i,d}^t|/L(i,d,t)\right). \tag{18}$$

And the probability density function of individual bat i is:

$$Q(x_i^t) = \prod_{d=1}^{D} \frac{1}{L(i,d,t)} \exp\left(-2|x_{i,d}^t - p_{i,d}^t|/L(i,d,t)\right). \tag{19}$$

Definition ζ_i^t is the probability measure corresponding to the D-dimensional double exponential distribution. Any Borel subset A of the set S satisfies $q[A] > 0$, so there are:

$$\zeta_i^t[A] = \int_A \left[\prod_{d=1}^{D} \frac{1}{L(i,d,t)} \exp\left(-2|x_{i,d}^t - p_{i,d}^t|/L(i,d,t)\right)\right] \mathrm{d}x_{i,1}^t \dots \mathrm{d}x_{i,D}^t, \tag{20}$$

$$K_i^t = \mathscr{R}^D \supset S, \tag{21}$$

where K_i^t represents the support of ζ_i^t on the sample space. So we get:

$$0 < \zeta_i^t[A] < 1. \tag{22}$$

The support union of individual bats in the solution space can be shown as:

$$K^t = \bigcup_{i=1}^{N} K_i^t, \tag{23}$$

$$= \mathscr{R}^D \supset S. \tag{24}$$

Among them, K^t represents the support of ζ^t, and the probability A generated by the distribution ζ^t can be recorded as:

$$\zeta_i^t[A] = 1 - \prod_{i=1}^{N} \left(1 - \zeta_i^t[A]\right). \tag{25}$$

From $0 < \zeta_i^t[A] < 1$, we can get:

$$0 < \zeta^t[A] < 1 \tag{26}$$

and there is:

$$\prod_{t=1}^{\infty} (1 - \zeta^t[A]) = 0 \tag{27}$$

Therefore, the CGQBA algorithm can satisfy Assumption 2. □

Lemma 3. The CGQBA algorithm is a globally convergent algorithm.

Proof. Because the CGQBA algorithm can satisfy Assumptions 1 and 2 at the same time, according to Theorem 1, it can be known that it is a globally convergent algorithm. □

Then, the optimization performance of the CGQBA algorithm will be tested through numerical simulation experiments.

5 NUMERICAL EXPERIMENT AND RESULT ANALYSIS

In this section, the optimization performance and convergence of the proposed CGQBA algorithm will be tested and verified through numerical simulation experiments. In order to ensure the objectivity and comprehensiveness of the comparison experiment, 20 standard test functions with different modes were selected for numerical experiments, including 7 single-peaked functions ($F_1 - F_7$), 6 multi-peaked functions ($F_8 - F_{13}$) and 7 fixed-dimensional functions ($F_{14} - F_{20}$), and the benchmark test functions are shown in Table 1. All numerical simulation experiments are tested using Matlab 2016a on PC Windows 10 with Intel Core i5-1035G1 CPU @ 1.00 GHz 1.19 GHz, 16 GB memory. In the numerical experiment part, three experiments are designed, which are the influence of the improvement strategy on the performance improvement of the algorithm, the comparison experiment with the four classic swarm intelligence algorithms, and the high-dimensional performance comparison with the QPSO algorithm, respectively.

5.1 Influence of the Introduced Strategies on the Performance of CGQBA Algorithm

In order to explore the impact of the two improved strategies designed on the performance of the CGQBA algorithm, the basic QBA algorithm is used as the basic control group to conduct ablation experiments with algorithms corresponding to different improved strategies in this subsection. Algorithms with different improvement strategies include: QBA algorithm with shrink factor strategy only (CQBA), QBA algorithm with adaptive golden sine mutation strategy only (GQBA), and QBA with both improvement strategies (CGQBA). In the ablative comparison experiments, two different population sizes and dimensionality settings were compared. First of all, under the setting of population size of 10, maximum iteration number of 500 and dimension of 2, the optimization results of the four algorithms running independently on 20 test functions 50 times were recorded. The comparison indicators include the fitness value and running time of the test function. The experimental results are shown in Table 2, where the best fitness value is marked in bold.

By analyzing the experimental results in Table 2, it can be seen that in terms of running time, the algorithm with the improved strategy has no obvious extra time loss compared with the QBA algorithm. The required running time difference

Function	Dimension	Optimal Value
$F_1(x) = \sum_{i=1}^{n} x_i^2$	D	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	D	0
$F_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	D	0
$F_4(x) = \max\{\lvert x_i \rvert, 1 \leq i \leq n\}$	D	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	D	0
$F_6(x) = \sum_{i=1}^{n} [(x_i + 0.5)^2]$	D	0
$F_7(x) = ix_i^4 + random[0, 1]$	D	0
$F_8(x) = -\sum_{i=1}^{n} \sin(\text{sqrt}\lvert x_i \rvert)$	D	$-418.982 \times n$
$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	D	0
$F_{10}(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sqrt{\sum_{i=1}^{n}\cos(2\pi x_i)}\right) + 20 + e$	D	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	D	0
$F_{12}(x) = \frac{\pi}{n} 10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_{i-1})^2 [1 + sin^2(\pi y_{i+1})] + (y_n - 1)^2$ $y_i = 1 + \frac{x_i+1}{4}$, $u = (x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a < x_i < a, \\ k(-x_i - a)^m, & x_i < a \end{cases}$	D	0
$F_{13}(x) = 0.1\{sin^2(3\pi x_i) + \sum_{i=1}^{n} (x_i - 1)^2 [1 + sin^2(3\pi x_i + 1)] + (x_i - 1)^2 [1 + \sin^2(2\pi x_i)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	D	0
$F_{14}(x) = (\frac{1}{500} + \sum_{i=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6})^{-1}$	2	1
$F_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_1({b_i}^2 + b_i x_2)}{{b_i}^2 + b_i x_3 + x_4}]^2$	4	0.00030
$F_{16}(x) = 4{x_1}^2 - 2.1{x_1}^4 + \frac{1}{3}{x_1}^6 + x_1 x_2 - 4{x_2}^2 - 4{x_2}^4$	2	-1.0316
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}{x_1}^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) cos x_1 + 10$	2	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] * [30 + (2x_1 + 3x_2)^2 * (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_i j (x_i - p_i j)^2)$	3	-3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_i j (x_i - p_i j)^2)$	6	-3.32

Table 1. Test function

Functions	Indicators	QBA	CQBA	GQBA	CGQBA
	Fitness	−4.57E−01	−4.33E−01	**0.00E+00**	**0.00E+00**
F_1	t	0.1399	0.1792	0.1505	0.1651
	T	5	4	6	13
	Fitness	−4.36E−01	−4.30E−01	5.52E−254	**0.00E+00**
F_2	t	0.1399	0.1667	0.1792	0.2005
	T	9	6	13	32
	Fitness	−4.40E−01	−4.32E−01	**0.00E+00**	**0.00E+00**
F_3	t	0.1505	0.1938	0.1880	0.1911
	T	6	4	7	14
	Fitness	−4.30E−01	−4.39E−01	1.15E−237	**0.00E+00**
F_4	t	0.1245	0.1766	0.1656	0.1771
	T	8	6	13	31
	Fitness	2.41E−01	1.52E−01	**6.80E−03**	2.82E−02
F_5	t	0.1318	0.1318	0.1307	0.1313
	T	485	457	500	500
	Fitness	−3.44E−01	−3.47E−01	1.67E−02	**1.43E−04**
F_6	t	0.1297	0.1672	0.1521	0.1677
	T	84	36	500	500
	Fitness	−4.15E−01	−4.04E−01	**1.84E−04**	1.85E−04
F_7	t	0.1333	0.1740	0.1690	0.1797
	T	18	17	500	500
	Fitness	−1.15E+01	−5.71E+01	−8.25E+01	**−4.19E+02**
F_8	t	0.0870	0.0964	0.0974	0.1146
	T	500	500	500	500
	Fitness	−4.24E−01	−4.24E−01	**0.00E+00**	**0.00E+00**
F_9	t	0.1016	0.1089	0.1052	0.1297
	T	8	6	10	22
	Fitness	−4.35E−01	−4.28E−01	8.88E−16	**8.88E−16**
F_{10}	t	0.1375	0.1792	0.1828	0.2073
	T	9	6	14	34
	Fitness	−4.29E−01	−4.31E−01	**0.00E+00**	**0.00E+00**
F_{11}	t	0.0969	0.1250	0.1120	0.1302
	T	6	5	16	23
	Fitness	−2.70E−01	−2.92E−01	3.54E−02	**3.70E−03**
F_{12}	t	0.1370	0.1437	0.1432	0.1745
	T	136	161	500	500
	Fitness	−3.09E−01	−3.36E−01	1.66E−02	**8.01E−04**
F_{13}	t	0.1495	0.1505	0.1484	0.1646
	T	86	69	500	500
	Fitness	1.22E+01	1.15E+01	1.27E+01	**7.88E+00**
F_{14}	t	0.4615	0.5437	0.5214	0.4854
	T	500	500	500	500

to be continued

Functions	Indicators	QBA	CQBA	GQBA	CGQBA
	Fitness	−4.07E−01	−4.10E−01	**3.75E−04**	1.00E−03
F_{15}	t	0.1406	0.1844	0.1672	0.1932
	T	26	20	500	500
	Fitness	−1.37E+00	−1.36E+00	−1.02E+00	**−1.03E+00**
F_{16}	t	0.1161	0.1599	0.1781	0.1786
	T	2	1	2	2
	Fitness	1.54E+00	3.68E−01	**4.00E−01**	4.17E−01
F_{17}	t	0.1187	0.1646	0.1510	0.1635
	T	500	495	500	500
	Fitness	8.37E+01	2.87E+01	3.99E+01	**3.01E+00**
F_{18}	t	0.1151	0.1734	0.1609	0.1708
	T	500	500	500	500
	Fitness	−1.98E+00	−3.04E+00	**−3.84E+00**	−3.70E+00
F_{19}	t	0.1573	0.2047	0.2047	0.2161
	T	500	500	500	500
	Fitness	−1.15E+00	−1.60E+00	**−3.18E+00**	−2.95E+00
F_{20}	t	0.1682	0.2120	0.1995	0.2208
	T	500	500	500	500

Table 2. Ablation comparison experimental results of the improved strategy ($D = 2$)

is approximately 0.15 s. In terms of convergence accuracy, the CQBA algorithm with only the shrink factor strategy has a slight improvement in the fitness value of the objective function compared with the QBA algorithm. The GQBA algorithm with the adaptive golden sine mutation strategy can significantly improve the convergence accuracy of QBA. For example, the theoretical optimal value of the test function is searched on F_1, F_3, F_9, and F_{11}. On F_2 and F_4, the convergence accuracy of the QBA algorithm is improved by more than 200 orders of magnitude. And there are different degrees of improvement in other test functions. Observation based on iteration numbers, the proposed algorithm requires more iterations, but the time consumption is not significantly increased, which reveals that the convergence rate of the improved algorithm is faster. For test function F_1, QBA runs 50 times independently, the number of iterations is about 6 times, the running time is 0.1399 s, and the search accuracy is −4.35E−01. While the number of iterations required by CGQBA is about 13 times, the running time is 0.1651 s, and the theoretical optimal value 0.00E+00 is found. Such experimental results show that the QBA may easily get stuck into local optimal values under certain conditions, just as mentioned in Section 1. While the CGQBA algorithm solves this defect effectively, the proposed algorithm can effectively avoid falling into the local extremum. In addition, the CGQBA algorithm with two improved strategies is the best among the four comparison algorithms. The results reflect that the improved algorithm combining the two strategies can synergistically promote the global exploration and local exploitation abilities of the algorithm.

In order to test the effect of changes in parameter settings on the performance of the improved strategy, the population size and dimension settings were both increased to 30, and the results are shown in Table 3. The comparison indicators are consistent with Table 2, which are obtained by running each algorithm independently 50 times and calculating the average value. It should be pointed out that F_{14}-F_{20} are test functions of fixed dimensions, and the numerical experiment of this part of the function only increases the population size to 30. By analyzing the numerical results, it can be seen that the algorithm with the improved strategy is better than the basic QBA algorithm with the increase of population size and dimension. And the CGQBA algorithm with two improved strategies still has the best optimization performance, and there is no situation that the optimization performance decreases with the increase of the dimension, which shows that the improved strategy is stable.

5.2 Performance Comparison with Other Algorithms

In order to further verify the optimization ability of the CGQBA algorithm, it is compared with the QBA algorithm [18], the particle swarm algorithm (PSO) [1], the golden sine algorithm (GSA) [23], and the Harris Hawks optimization algorithm (HHO) [25].

The basic parameters such as the maximum number of iterations, population size, and dimension of each algorithm are set the same in order to ensure the fairness of the experiments. The characteristic parameter settings of the comparison algorithms are shown in Table 4. The comparative results of the five algorithms are shown in Table 5. The results of searching for the theoretical optimal value are shown in bold. The optimal value and standard deviation of the objective function in the table can reflect the convergence accuracy and optimization ability of the algorithm. From the numerical experiment results, it can be intuitively concluded that the basic QBA algorithm does not search for the theoretical optimal solution on all test functions. The PSO algorithm only converges to the theoretical optimal solution on the 4 test functions F_{16} to F_{19}. The GSA algorithm only converges to the theoretical optimal solution on the 4 test functions F_9, F_{11}, F_{17} and F_{18}. The HHO algorithm only converges to the theoretical optimal solution on the 6 test functions F_9, F_{11}, $F_{17} - F_{19}$. The CGQBA algorithm proposed in this paper converges to the theoretical optimal solution on 12 test functions $F_1 - F_4$, F_9, F_{11} and $F_{15} - F_{20}$, and the optimization performance is better. Through the solution results of five algorithms on 20 test functions, it can be clearly concluded that the CGQBA algorithm has significantly better optimization performance when dealing with test functions of different modes. By comparing with other algorithms, the CGQBA algorithm obtains higher optimization accuracy in the solution process.

5.3 High-Dimensional Performance Comparison with QPSO Algorithm

In order to explore the optimization performance of the CGQBA algorithm in high-dimensional settings, the proposed algorithm is compared with the classical QPSO

Functions	Indicators	QBA	CQBA	GQBA	CGQBA
	Fitness	−4.66E−01	−4.71E−01	**0.00E+00**	**0.00E+00**
F_1	t	0.3781	0.4609	0.5182	0.5724
	T	9	4	9	23
	Fitness	−4.70E−01	−4.58E−01	2.28E−236	**0.00E+00**
F_2	t	0.3943	0.5578	0.5625	0.5630
	T	11	9	18	49
	Fitness	−4.57E−01	−4.69E−01	**0.00E+00**	**0.00E+00**
F_3	t	1.2615	1.3859	1.4193	1.3714
	T	8	5	10	28
	Fitness	−4.61E−01	−4.63E−01	1.33E-201	**0.00E+00**
F_4	t	0.3870	0.5271	0.5188	0.5385
	T	10	7	17	47
	Fitness	2.85E+01	2.84E+01	**2.75E+01**	2.83E+01
F_5	t	0.3693	0.4167	0.3698	0.4234
	T	500	500	500	500
	Fitness	5.42E+00	3.13E+00	4.02E+00	**1.98E+00**
F_6	t	0.3896	0.5198	0.5089	0.5021
	T	500	500	500	500
	Fitness	−4.46E−01	−4.49E−01	**6.33E−05**	9.33E−05
F_7	t	0.5318	0.6375	0.6411	0.6604
	T	15	15	500	500
	Fitness	−5.71E+01	−3.28E+02	−3.14E+02	**−1.19E+03**
F_8	t	0.2995	0.3214	0.3292	0.3026
	T	500	500	500	500
	Fitness	−4.67E−01	−4.62E−01	**0.00E+00**	**0.00E+00**
F_9	t	0.3474	0.3589	0.3583	0.4026
	T	10	6	13	33
	Fitness	−4.78E−01	−4.57E−01	8.88E−16	**8.88E−16**
F_{10}	t	0.4552	0.5370	0.5677	0.4807
	T	10	7	17	47
	Fitness	−4.71E−01	−4.72E−01	**0.00E+00**	**0.00E+00**
F_{11}	t	0.3417	0.3859	0.3792	0.3740
	T	6	4	8	18
	Fitness	4.90E−01	−1.21E−01	5.05E−01	**9.00E−02**
F_{12}	t	0.6792	0.6927	0.7719	0.6526
	T	500	311	500	500
	Fitness	2.33E+00	1.55E+00	3.12E−01	**7.49E−02**
F_{13}	t	0.6474	0.7078	0.6979	0.6979
	T	500	500	500	500
	Fitness	1.19E+01	1.10E+01	1.27E+01	6.40E+00
F_{14}	t	**1.4052**	1.5089	1.5167	1.5682
	T	500	500	500	500

to be continued

Functions	Indicators	QBA	CQBA	GQBA	CGQBA
	Fitness	−4.49E−01	−4.51E−01	**3.46E−04**	7.37E−04
F_{15}	t	0.3870	0.5109	0.5260	0.5333
	T	16	9	500	500
	Fitness	−1.38E+00	−1.43E+00	−1.03E+00	**−1.03E+00**
F_{16}	t	0.3682	0.4573	0.4453	0.5203
	T	500	500	500	500
	Fitness	2.29E−01	1.86E−01	**3.99E−01**	4.07E−01
F_{17}	t	0.3411	0.4474	0.4349	0.5245
	T	500	500	500	500
	Fitness	5.59E+00	5.01E+00	2.60E+01	**3.01E+00**
F_{18}	t	0.3448	0.4792	0.4516	0.4906
	T	500	500	500	500
	Fitness	−3.05E+00	−3.32E+00	**−3.86E+00**	−3.82E+00
F_{19}	t	0.4641	0.5885	0.6036	0.6125
	T	500	500	500	500
	Fitness	−1.70E+00	−2.25E+00	**−3.25E+00**	−3.10E+00
F_{20}	t	0.4458	0.5719	0.5604	0.6266
	T	500	500	500	500

Table 3. Ablation comparison experimental results of the improved strategy ($D = 30$)

Algorithms	Parameters
CGQBA	$A_{min} = 0$, $A_{max} = 0.85$, $r_{min} = 0$, $r_{max} = 0.8$
QBA	$A_{min} = 0$, $A_{max} = 0.85$, $r_{min} = 0$, $r_{ma}x = 0.8$
PSO	$C1 = C2 = 2$, $\omega_{max} = 0.9$, $\omega_{min} = 0.2$
GSA	$a = -\pi$, $b = \pi$
HHO	/

Table 4. The related parameters setting

algorithm in this subsection. For the first 12 test functions in Table 1 that can perform dimension expansion, 100-dimensional and 200-dimensional high-dimensional test experiments are carried out respectively. Each algorithm runs 50 times independently under each dimension setting, and calculates the test function fitness value and iterations to examine the optimization performance of the algorithm. For each index, the average value of 50 test results is calculated, and the results are shown in Table 6, in which the best value is shown in bold.

By analyzing the results in Table 6, it can be seen that the proposed CGQBA algorithm can effectively solve high-dimensional problems and exhibit good stability. In terms of the number of iterations, the CGQBA algorithm performs better in most test functions in different dimensions. The comparison of the specific iterations of the two algorithms under different high-dimensional settings is shown in Figure 3. It can be clearly seen that the number of iterations required by the CGQBA algorithm on 8 test functions ($F_1 - F_4$, $F_8 - F_{11}$) are significantly better than that of the QPSO

Functions	Indicators	QBA	PSO	GSA	HHO	CGQBA
F_1	Fitness	−4.66E−01	1.29E−05	1.01E−249	1.30E−97	**0.00E+00**
	Std.	5.55E−01	2.18E−05	0.00E+00	6.05E−97	2.20E−03
F_2	Fitness	−4.70E−01	7.44E−03	3.04E−130	5.94E−52	**0.00E+00**
	Std.	5.94E−01	9.92E−03	1.64E−129	1.56E-51	1.44E−01
F_3	Fitness	−4.57E−01	3.25E+01	1.22E−232	9.61E-70	**0.00E+00**
	Std.	1.20E−01	8.28E+01	0.00E+00	5.17E−69	7.73E−02
F_4	Fitness	−4.61E−01	4.82E−01	5.69E-111	2.24E−48	**0.00E+00**
	Std.	1.27E−01	4.82E−01	3.06E-110	1.05E−47	1.08E−02
F_5	Fitness	2.85E+01	4.79E+01	4.24E−03	1.64E−02	2.83E+01
	Std.	1.15E−01	3.66E+01	7.43E−03	2.64E−02	6.39E−02
F_6	Fitness	5.42E+00	8.25E−06	3.91E−04	1.02E−04	1.98E+00
	Std.	1.88E−01	1.58E−05	9.23E−04	1.03E−04	6.18E−01
F_7	Fitness	−4.46E−01	8.97E−02	1.19E−04	1.03E−04	9.33E−05
	Std.	3.76E+001	8.97E−02	1.59E−04	7.45E−05	4.83E−02
F_8	Fitness	−5.71E+01	4.66E+01	−1.26E+04	−1.26E+04	−1.90E+03
	Std.	9.19E+00	5.00E+02	1.98E−01	1.26E+00	6.09E+01
F_9	Fitness	−4.67E−01	1.29E−03	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std.	1.59E+00	1.17E+01	0.00E+00	0.00E+00	5.90E−03
F_{10}	Fitness	−4.78E−01	4.42E+01	8.88E−16	8.88E−16	8.88E−16
	Std.	1.32E−01	8.60E−04	9.86E−32	9.86E−32	2.50E−03
F_{11}	Fitness	−4.71E−01	4.42E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std.	1.42E+00	7.08E+00	0.00E+00	0.00E+00	2.98E−04
F_{12}	Fitness	4.90E−01	7.87E−01	1.39E−05	5.88E−06	9.00E−02
	Std.	1.17E−01	9.35E−01	2.23E−05	8.88E−06	1.76E−02
F_{13}	Fitness	2.33E+00	3.30E−03	3.39E−05	1.55E−04	7.49E−02
	Std.	1.12E−01	5.03E−03	9.09E−05	2.66E−04	6.97E−02
F_{14}	Fitness	1.19E+01	1.59E+00	1.10E+00	1.39E+00	6.40E+00
	Std.	8.02E−01	1.36E+00	3.84E−01	9.40E−01	6.89E−04
F_{15}	Fitness	−4.49E−01	5.42E−04	5.33E−04	3.51E−04	**3.00E−04**
	Std.	1.36E−01	3.74E−04	4.49E−04	3.95E−05	8.94E−04
F_{16}	Fitness	−1.38E+00	**−1.03E+00**	−1.02E+00	**−1.03E+00**	**−1.03E+00**
	Std.	1.84E−01	0.00E+00	8.98E−03	4.15E−09	1.25E−04
F_{17}	Fitness	2.29E−01	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**
	Std.	1.47E−01	1.11E−16	8.22E−04	1.69E−05	3.63E−15
F_{18}	Fitness	5.59E+00	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**
	Std.	5.05E+00	1.92E−15	9.64E+00	9.81E−07	4.70E−01
F_{19}	Fitness	−3.05E+00	**−3.86E+00**	−3.80E+00	**−3.86E+00**	**−3.86E+00**
	Std.	1.02E−01	2.66E−15	6.69E−02	2.02E−03	1.22E−01
F_{20}	Fitness	−1.70E+00	−3.28E+00	−2.98E+00	−3.12E+00	**−3.30E+00**
	Std.	2.31E−01	5.60E−02	3.30E−01	8.61E−02	4.59E−01

Table 5. Comparison results of CGQBA and the same type of algorithms ($N = 30$, $D = 30$)

N, D	Functions	Algorithms	Fitness	T
20, 100	F_1	CGQBA	**0.00E+00**	**22**
		QPSO	3.61E−02	500
	F_2	CGQBA	**0.00E+00**	**41**
		QPSO	3.69E−01	500
	F_3	CGQBA	**0.00E+00**	**30**
		QPSO	4.49E+03	500
	F_4	CGQBA	**0.00E+00**	**51**
		QPSO	2.04E+01	500
	F_5	CGQBA	**9.85E+01**	500
		QPSO	9.74E+02	500
	F_6	CGQBA	**1.27E+01**	500
		QPSO	2.07E+01	500
	F_7	CGQBA	**1.07E+04**	500
		QPSO	4.05E+02	500
	F_8	CGQBA	**−3.54E+03**	**50**
		QPSO	−7.61E+02	500
	F_9	CGQBA	**0.00E+00**	**30**
		QPSO	1.63E+02	500
	F_{10}	CGQBA	**8.88E−16**	**40**
		QPSO	6.52E−02	500
	F_{11}	CGQBA	**0.00E+00**	**18**
		QPSO	1.61E−02	500
	F_{12}	CGQBA	**2.09E−01**	500
		QPSO	2.49E+02	500

to be continued

algorithm. The QPSO algorithm has not converged when it reaches the maximum number of iterations on all tested functions. In addition, with the large increase of dimensions, the performance of the CGQBA algorithm proposed in this paper does not decrease, and it has good stability and robustness. In contrast to the QPSO algorithm, with the substantial increase of the dimension, the optimization performance of the algorithm decreases significantly. In terms of convergence accuracy, it can be obtained by analyzing the experimental data that the convergence accuracy of CGQBA algorithm on all test functions is higher than that of QPSO algorithm, and the CGQBA algorithm has searched for the theoretical optimal value on all functions.

For the purpose of further evaluating the solution performance of the improved CGQBA algorithm, the Wilcoxon signed-rank test was performed on the best results of 50 independent runs of the CGQBA algorithm and the QPSO algorithm under the 100-dimensional and 200-dimensional settings, respectively. And the significance level is set to be 5%. The symbols "+", "-" and "=" respectively indicate that the performance of the CGQBA algorithm is better than, worse than and equivalent to the QPSO algorithm. The results of the Wilcoxon signed-rank test are shown

N, D	Functions	Algorithms	Fitness	T
20, 200	F_1	CGQBA	**0.00E+00**	**24**
		QPSO	2.27E+02	500
	F_2	CGQBA	**0.00E+00**	**44**
		QPSO	2.26E+01	500
	F_3	CGQBA	**0.00E+00**	**35**
		QPSO	7.30E+04	500
	F_4	CGQBA	**0.00E+00**	**58**
		QPSO	2.86E+01	500
	F_5	CGQBA	**1.98E+02**	500
		QPSO	8.65E+07	500
	F_6	CGQBA	**2.69E+01**	500
		QPSO	3.40E+02	500
	F_7	CGQBA	**1.14E−04**	500
		QPSO	6.63E+07	500
	F_8	CGQBA	**−8.53E+03**	**50**
		QPSO	−1.03E+03	500
	F_9	CGQBA	**0.00E+00**	**30**
		QPSO	1.79E+03	500
	F_{10}	CGQBA	**8.88E−16**	**43**
		QPSO	4.89E+00	500
	F_{11}	CGQBA	**0.00E+00**	**17**
		QPSO	9.78E−01	500
	F_{12}	CGQBA	**2.27E−01**	500
		QPSO	4.35E+06	500

Table 6. Performance comparison of 12 benchmarks with high dimensions

Functions	CGQBA vs. QPSO p-value (100 D)	CGQBA vs. QPSO R	CGQBA vs. QPSO p-value (200 D)	CGQBA vs. QPSO R
F_1	.000	+	.000	+
F_2	.000	+	.000	+
F_3	.000	+	.000	+
F_4	.000	+	.000	+
F_5	.000	+	.000	+
F_6	.000	+	.000	+
F_7	.000	+	.000	+
F_8	.000	+	.000	+
F_9	.000	+	.000	+
F_{10}	.000	+	.000	+
F_{11}	.000	+	.000	+
F_{12}	.144	−	.000	+

Table 7. P-value of numerical simulation results under Wilcoxon rank-sum test

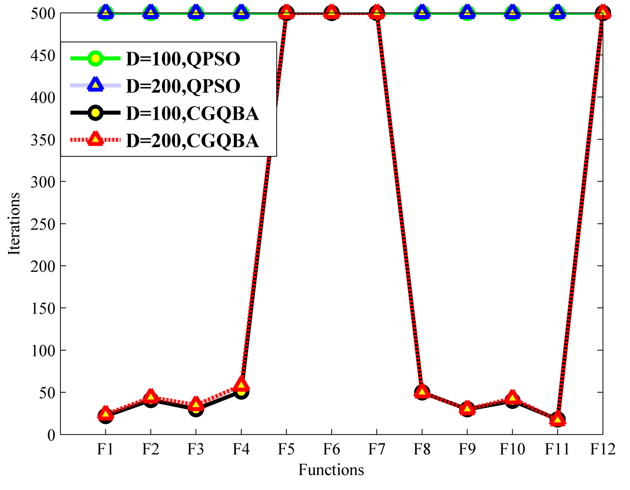


Figure 3. Comparison of optimization iteration times for CGQBA and QPSO algorithm under different dimensions

in Table 7, and it illustrates that the performance of CGQBA algorithm is better than QPSO algorithm on 11 functions when the dimension is 100 dimensions, and the p-values of CGQBA are all less than 0.01. When the dimension is 200, the solution performance of CGQBA algorithm is still stable and all better than that of the QPSO algorithm. Such statistical test results show that the superiority of the CGQBA algorithm is statistically significant.

In view of the convergence of the two algorithms under different dimensions, the dimension setting was increased from 50 dimensions to 100 dimensions, 150 dimensions and 200 dimensions in turn. And the convergence iteration curves of the two algorithms for the test function under different dimension settings were drawn respectively. An intuitive comparison of the convergence performance of the CGQBA and the QPSO algorithm is shown in Figure 4. In these convergence plots, the horizontal and vertical axes represent the number of iterations and fitness values, respectively. It can be clearly revealed that under the same dimension setting, the convergence performance of the proposed CGQBA algorithm on all test functions is significantly better than that of the QPSO algorithm. Whether it is to solve a unimodal function or a multimodal function, the convergence accuracy of the CGQBA algorithm is higher, and the number of iterations required to achieve convergence is significantly less than that of the QPSO. In addition, with the large increase of dimension setting, the difficulty of testing functions increases rapidly. The optimization performance of the CGQBA algorithm proposed is more stable than that of the QPSO algorithm. For example, the CGQBA algorithm has searched for the theoretical optimal solution in less than 100 iterations as the dimension setting continues to increase for test functions F_9, F_{10} and F_{11}. In contrast to the QPSO algorithm, with the increase of the dimension, the optimization performance of the algorithm decreases.

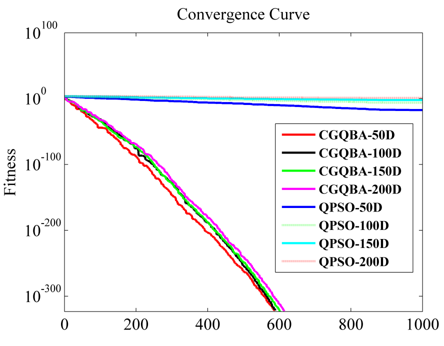
From the results of the above numerical simulation experiments, it can be concluded that compared with the QPSO algorithm, the CGQBA algorithm can still quickly converge to the optimal value in almost all test functions, and is not affected by the drastic changes in the data dimension. With the sharp increase of the dimension, the CGQBA algorithm shows stronger convergence ability than the QPSO algorithm at 100 and 200 dimensions. The comparison results show that the CGQBA algorithm has obvious advantages in processing high-dimensional complex data. Next, the multi-threshold image segmentation problem has been transformed into an optimization problem, and the proposed improved algorithm is applied to it to further verify its practicability in real-world optimization.

6 APPLICATION OF IMPROVED ALGORITHM IN MULTI-THRESHOLD SEGMENTATION

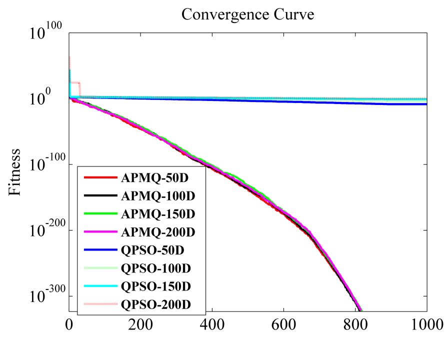
Image segmentation is one of the important methods of image processing, and it is a preprocessing step for effective analysis and understanding of images. The maximum between-class variance method is an effective image segmentation method that adaptively determines the threshold value. It has been favored by scholars because of its stable segmentation performance. However, with the expansion of the threshold dimension, the computational complexity of the corresponding segmentation problem grows exponentially, making it very tough to solve. Since most problems in the image processing direction can be transformed into optimization problems, swarm intelligence algorithm has been successfully applied in this field as an effective method to solve optimization problems. Nevertheless there is not much research literature on solving image segmentation problems using BA algorithms with quantum behavior. Therefore, using the proposed CGQBA algorithm to solve the image segmentation problem can not only provide a new solution idea for the corresponding image segmentation problem, but also further expand the application scope of the algorithm. This section combines the multi-threshold segmentation method with the CGQBA algorithm to construct a new image segmentation model, in order to verify the effectiveness and practicability of the proposed algorithm.

6.1 Multi-Threshold Image Segmentation Model Based on CGQBA Algorithm

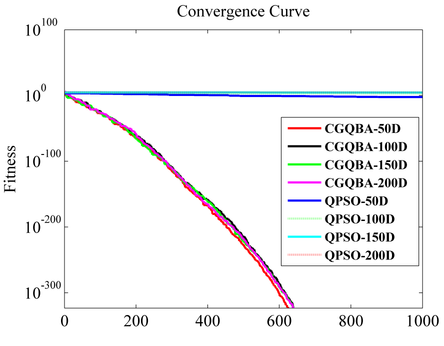
The maximum between-class variance method (Otsu) was proposed by the Japanese scholar Otsu. It divides the image into two categories: background and foreground according to the grayscale characteristics of the image to be processed, and then it uses variance to segment the image. This method is not affected by the characteristics of image brightness and contrast, it is a commonly used method for a class of



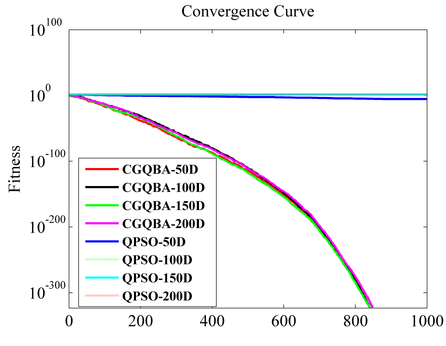
a) F_1



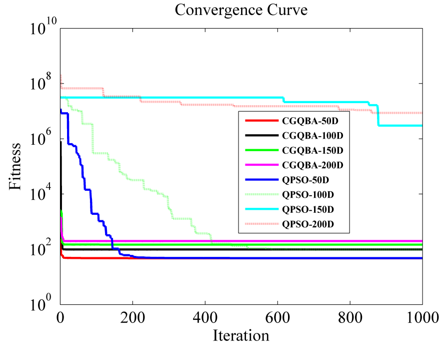
b) F_2



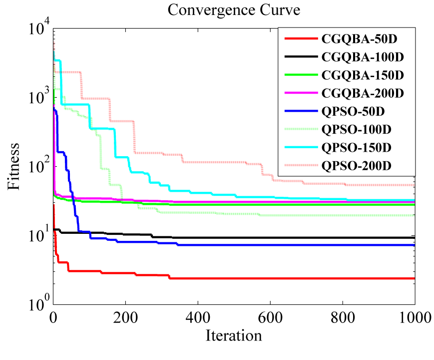
c) F_3



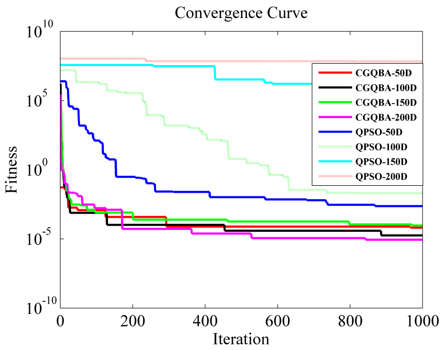
d) F_4



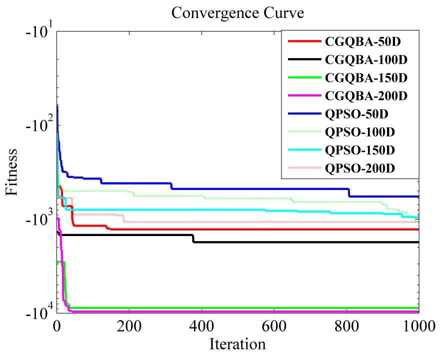
e) F_5



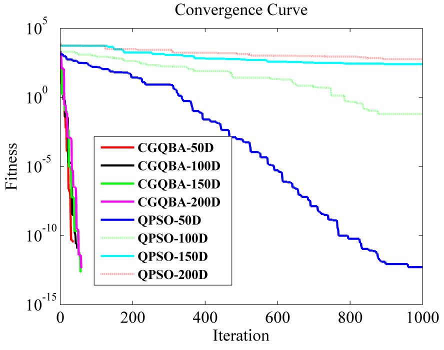
f) F_6



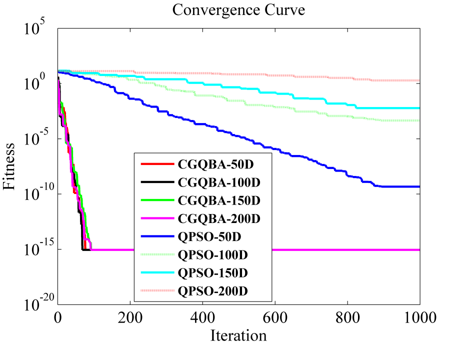
g) F_7



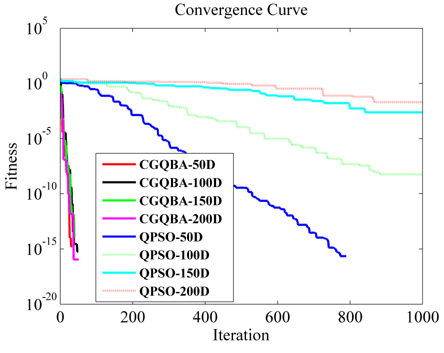
h) F_8



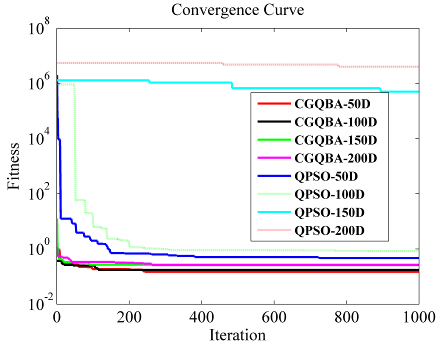
i) F_9



j) F_{10}



k) F_{11}



l) F_{12}

Figure 4. Comparison of convergence between CGQBA and QPSO in different dimensions

image segmentation. The standard Otsu method is a single-threshold segmentation method. Assuming that the gray levels of the image to be segmented are expressed as 0, 1, ..., $L-1$, respectively, then the pixels with gray level j can be expressed as $m_j = hist(j)$. The single threshold t divides the image into 2 regions, and the average probability level of each region is expressed as:

$$c_0 = \sum_{j=0}^{t} \frac{j \cdot hist(j)}{P_0}, \tag{28}$$

$$c_1 = \sum_{j=t}^{L-1} \frac{j \cdot hist(j)}{P_1}, \tag{29}$$

where P_0 and P_1 represent the probability distribution of the two divided regions respectively, and their specific expressions are:

$$P_0 = \sum_{j=0}^{t-1} hist(j), \tag{30}$$

$$P_1 = \sum_{j=t}^{L-1} hist(j). \tag{31}$$

The between-class variance of the single-threshold Otsu method is recorded as:

$$S_b^* = P_0(c_0 - E_g)^2 + P_1(c_1 - E_g)^2, \tag{32}$$

where E_g is the expected gray value of the image to be segmented. The single-threshold Otsu method can be extended to multiple thresholds, assuming that the n thresholds for segmentation are: $TH = [t_1, t_2, \ldots, t_n]$, and the n thresholds divide the image into C_0, C_1, ..., C_n classes, and the gray levels corresponding to these $n+1$ classes are respectively expressed as: 0, 1, 2, ..., t_1, t_{1+1}, t_{1+2}, ..., t_2, t_{2+1}, t_{2+2}, ..., t_n, t_{n+1}, t_{n+2}, ..., $L-1$, so the inter-class variance of the multi-threshold Otsu method can be written as:

$$S_b = \sum_{j=0}^{n} P_j(m_j - E_g)^2. \tag{33}$$

Among them, the average probability level c_j and probability distribution P_j of each divided region can be expressed as:

$$\begin{cases} c_0 = \sum_{j=0}^{t_1-1} \dfrac{j \cdot hist(j)}{P_0}, \\ c_j = \sum_{j=t_i}^{t_{j+1}-1} \dfrac{j \cdot hist(j)}{P_j}, \\ \vdots \\ c_{n-1} = \sum_{j=t_n}^{L-1} \dfrac{j \cdot hist(j)}{P_{n-1}}, \end{cases} \tag{34}$$

$$\begin{cases} P_0 = \sum_{j=0}^{t_1-1} hist(j), \\ P_j = \sum_{j=t_j}^{t_{j+1}-1} hist(j), \\ \vdots \\ P_{n-1} = \sum_{j=t_n}^{L-1} hist(j). \end{cases} \tag{35}$$

The objective function of the image segmentation problem based on Otsu method can be expressed as:

$$F(TH) = S_b. \tag{36}$$

And the optimization objective function of the multi-threshold maximum inter-class variance method based on the CGQBA algorithm can be expressed as:

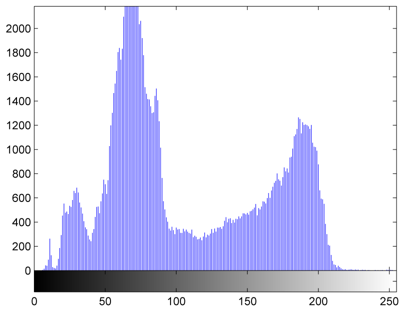
$$\{t_1, t_2, \ldots, t_n\} = \arg\max\{F(TH)\}. \tag{37}$$

So far, the multi-threshold image segmentation problem has been transformed into an optimization problem based on threshold selection. Next, this optimization problem will be solved by using the proposed CGQBA algorithm.

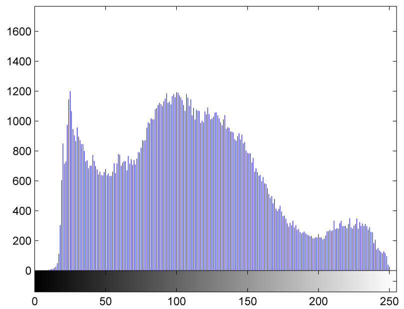
6.2 Segmentation Experiment Results and Analysis

In this subsection, the effectiveness and practicability of the multi-threshold image segmentation based on the CGQBA algorithm are verified, and they are compared with three traditional image segmentation methods. Six test images in the classic database BSDS300 are selected, and the histogram of the test images is shown in Figure 5. The parameters of the CGQBA algorithm are set as: the maximum number of iterations is 100, and the population size is 20.

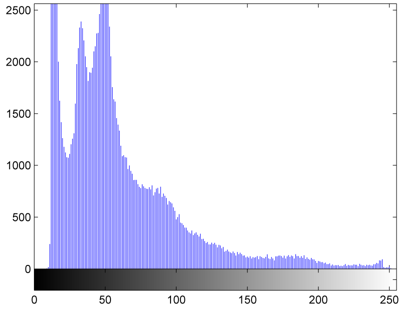
In order to verify the effectiveness of the improved CGQBA algorithm to solve the segmentation problem, a segmentation comparison experiment was conducted on 6 test images with three classic segmentation algorithms, namely the watershed segmentation algorithm, the local threshold algorithm and the region growing



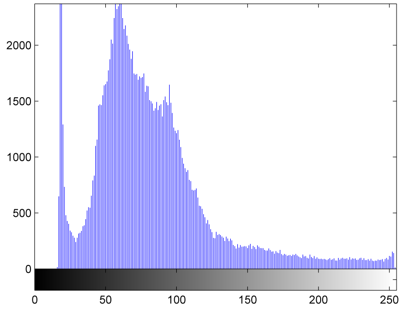
a) image 1



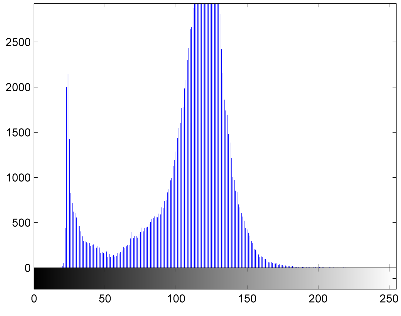
b) image 2



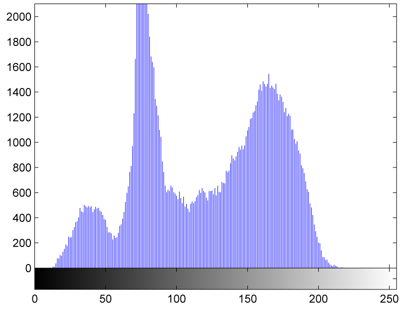
c) image 3



d) image 4



e) image 5



f) image 6

Figure 5. Histogram of 6 test images

algorithm. The segmentation results are shown in four groups of subgraphs in Figures 6 a), 6 c), 6 e), and 6 g). It can be clearly illustrated from the segmentation results that the proposed CGQBA algorithm can effectively solve the image segmentation problem, and its segmentation results retain more detailed information. For example, under noise-free conditions, the segmentation results of the six test images by the watershed algorithm all contain redundant area lines, which reduces the segmentation accuracy. The local threshold algorithm has the misclassification of background and foreground in the test image, especially the misclassification of test images 1, 4, and 5 is more serious. Except for test images 2 and 6, the segmentation results of the watershed algorithm are worse than other algorithms. The segmentation results of the proposed CGQBA algorithm are rich in detail and more stable in all test images.

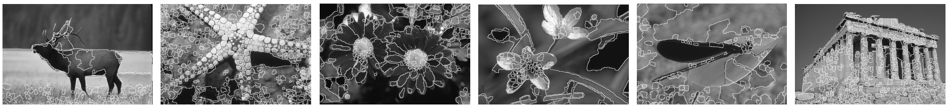
In addition, in order to increase the difficulty of segmentation and further test the performance of the algorithm, Gaussian noise was added to all test images to test the anti-noise ability of each contrasting algorithm. The results of this part of the segmentation are shown in four groups of subgraphs in Figures 6 b), 6 d), 6 f), and 6 h). After adding Gaussian noise to the test image, the segmentation performance of several traditional algorithms is significantly reduced. It can be clearly observed from the results that the segmentation accuracy is significantly reduced. On the other hand, the segmentation effect of the CGQBA algorithm is still better under the interference of Gaussian noise, which reflects the algorithm has anti-noise ability and better stability.

It can be seen from the above results that the proposed CGQBA algorithm effectively improves the convergence speed and accuracy, and can effectively solve the real-world optimization of multi-threshold segmentation problem. Its segmentation accuracy and ability to deal with noise interference are better than the traditional segmentation algorithms compared.

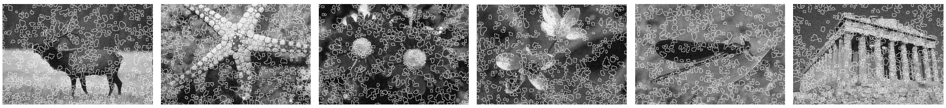
7 CONCLUSION

Aiming at the shortcomings of the basic QBA algorithm that the optimization accuracy is reduced in the later search stage and it is easy to fall into the local extrema, an improved QBA algorithm (CGQBA) based on the directional convergence factor and the self-learning predictive mutation mechanism is proposed. Numerical simulation and comparison experiments with various algorithms on 20 test functions with different characteristics show that the CGQBA algorithm is superior to the basic QBA algorithm and other comparison algorithms. In addition, in the practical application of image segmentation problem, the CGQBA algorithm also shows better segmentation performance and anti-noise ability.

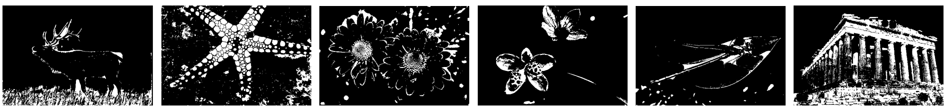
Despite the good performance, it is worth pointing out that the results obtained are preliminary. Further studies can validate the proposed algorithm further by a systematic tuning of relevant parameters and the test of even higher-dimensional problems. In the follow-up research, we will continue to optimize the algorithm



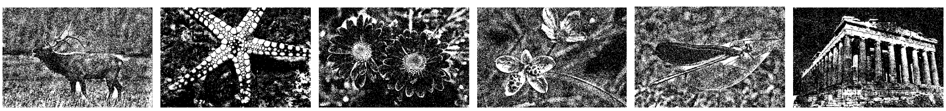
a) Segmentation results without noise



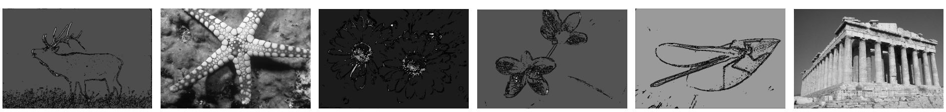
b) Segmentation results with Gaussian noise; Segmentation results of the Watershed Algorithm



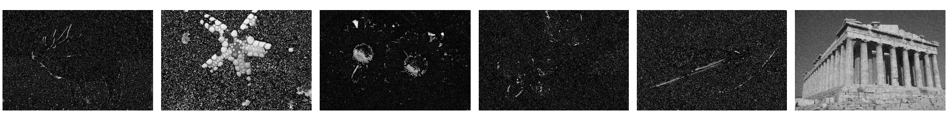
c) Segmentation results without noise



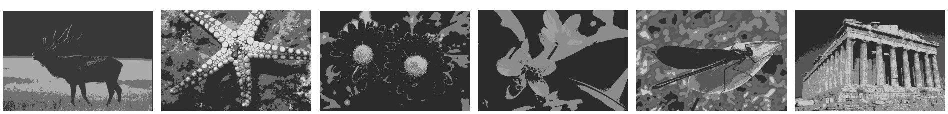
d) Segmentation results with Gaussian noise; Segmentation results of the Local Thresholding Algorithm



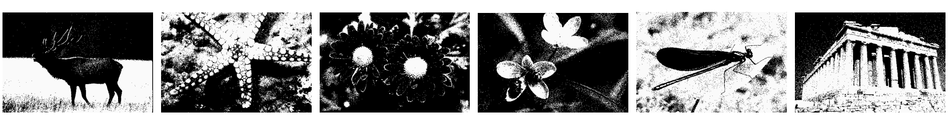
e) Segmentation results without noise



f) Segmentation results with Gaussian noise; Segmentation results of the Region Growing Algorithm



g) Segmentation results without noise



h) Segmentation results with Gaussian noise; Segmentation results of CGQBA Algorithm

Figure 6. Segmentation comparison between CGQBA and traditional algorithms

to ensure the optimization accuracy while reducing the time consumption of the algorithm. And the improved quantum behavioral bat algorithm can be applied to the problem of abnormal brain image segmentation to further verify the performance of the algorithm in complex problems.

8 COMPETING INTERESTS

The authors declare that they have no competing interests.

Acknowledgments

This work is supported by the Scientific Research Program Funded by Education Department of Shaanxi Provincial Government (Program No. 21JK0962), and the Natural Science Basic Research Project of Shaanxi Province, China (Grant No. 2024JC-YBMS-064).

REFERENCES

[1] Eberhart, R.—Kennedy, J.: A New Optimizer Using Particle Swarm Theory. Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS '95), 1995, pp. 39–43, doi: 10.1109/MHS.1995.494215.

[2] Dorigo, M.—Maniezzo, V.—Colorni, A.: Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems Man and Cybernetics, Part B (Cybernetics), Vol. 26, 1996, No. 1, pp. 29–41, doi: 10.1109/3477.484436.

[3] Passino, K. M.: Biomimicry of Bacterial Foraging for Distributed Optimization and Control. IEEE Control Systems Magazine, Vol. 22, 2002, No. 3, pp. 52–67, doi: 10.1109/MCS.2002.1004010.

[4] Karaboga, D.—Basturk, B.: A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. Journal of Global Optimization, Vol. 39, 2007, No. 3, pp. 459–471, doi: 10.1007/s10898-007-9149-x.

[5] Yang, X. S.: A New Metaheuristic Bat-Inspired Algorithm. Springer, 2010, doi: 10.1007/978-3-642-12538-6_6.

[6] Gandomi, A. H.—Yang, X. S.—Alavi, A. H.: Cuckoo Search Algorithm: A Metaheuristic Approach to Solve Structural Optimization Problems. Engineering with Computers, Vol. 29, 2013, No. 1, pp. 17–35, doi: 10.1007/s00366-011-0241-y.

[7] Mirjalili, S.—Mirjalili, S. M.—Lewis, A.: Grey Wolf Optimizer. Advances in Engineering Software, Vol. 69, 2014, pp. 46–61, doi: 10.1016/j.advengsoft.2013.12.007.

[8] Mirjalili, S.—Lewis, A.: The Whale Optimization Algorithm. Advances in Engineering Software, Vol. 95, 2016, pp. 51–67, doi: 10.1016/j.advengsoft.2016.01.008.

[9] Xue, J.—Shen, B.: A Novel Swarm Intelligence Optimization Approach: Sparrow Search Algorithm. Systems Science and Control Engineering, Vol. 8, 2020, No. 1, pp. 22–34, doi: 10.1080/21642583.2019.1708830.

[10] Long, W.—Zhang, W.: Modified Bat Algorithm for Solving Constrained Optimization Problems. Applied Research of Computers, Vol. 31, 2014, No. 8, pp. 2350–2353, doi: 10.3969/j.issn.1001-3695.2014.08.026 (in Chinese).

[11] Dai, H.—Luo, Y.: Wireless Network Traffic Prediction Based on Bat Algorithm Optimized Back Propagation Neural Network Model. Journal of Computer Applications, Vol. 41, 2021, No. S1, pp. 185–188, doi: 10.11772/j.issn.1001-9081.2020101679 (in Chinese).

[12] Li, C. W.—Gan, Y.—Sun, F. J.—Zhang, P. J.: Research on Robot Path Planning Based on Bat Algorithm and Artificial Potential Field. Manufacturing Automation, Vol. 43, 2021, No. 2, pp. 76–81 (in Chinese).

[13] Rauf, H. T.—Gao, J.—Almadhor, A.—Arif, M.—Nafis, M. T.: Enhanced Bat Algorithm for COVID-19 Short-Term Forecasting Using Optimized LSTM. Soft Computing, Vol. 25, 2021, No. 20, pp. 12989–12999, doi: 10.1007/s00500-021-06075-8.

[14] Li, L. L.—Zhou, W. N.—Liu, S. Y.: Target Tracking Based on DE Bat Algorithm for Particle Filter Optimization. Computer Systems and Applications, Vol. 28, 2019, No. 2, pp. 24–32, doi: 10.15888/j.cnki.csa.006759 (in Chinese).

[15] Chen, Z. M.—Tian, M. C.—Wu, P. L.—Bo, Y. M.—Gu, F. F.—Yue, C.: Intelligent Particle Filter Based on Bat Algorithm. Acta Physica Sinica, Vol. 66, 2017, No. 5, pp. 47–56 (in Chinese).

[16] Li, Z. J.: Improved Bat Algorithm Based on Sobol Sequence and Intermittent Lévy Jumping. Mathematics in Practice and Theory, Vol. 51, 2021, No. 8, pp. 313–320 (in Chinese).

[17] Li, Z. J.: Improved Bat Algorithm Based on Grouping Evolution and Hybrid Optimization. Mathematics in Practice and Theory, Vol. 50, 2020, No. 24, pp. 141–149 (in Chinese).

[18] Li, Z. Y.—Ma, L.—Zhang, H. Z.: A Quantum-Behaved Bat Algorithm for Integer Programming. Computer Engineering and Science, Vol. 36, 2014, No. 7, pp. 1336–1340 (in Chinese).

[19] Zhu, B.—Zhu, W.—Liu, Z.—Duan, Q.—Cao, L.: A Novel Quantum-Behaved Bat Algorithm with Mean Best Position Directed for Numerical Optimization. Computational Intelligence and Neuroscience, Vol. 2016, 2016, Art. No. 6097484, doi: 10.1155/2016/6097484.

[20] Rugema, F. X.—Yan, G.—Mugemanyi, S.—Jia, Q.—Zhang, S.—Bananeza, C.: A Cauchy-Gaussian Quantum-Behaved Bat Algorithm Applied to Solve the Economic Load Dispatch Problem. IEEE Access, Vol. 9, 2021, pp. 3207–3228, doi: 10.1109/ACCESS.2020.3034730.

[21] Yang, W.—Chen, G.—Su, J.: Quantum-Behaved Bat Algorithm for Solving Collaborative Scheduling Optimization of Multi-Stacker. Journal of Electronic Measurement and Instrumentation, Vol. 32, 2018, No. 4, pp. 128–133, doi: 10.13382/j.jemi.2018.04.018 (in Chinese).

[22] Gao, H.—Du, Y.—Zhang, S.—Liu, D.: Optimal Cooperative Strategy Based on Quantum Bat Algorithm for Cognitive Radio of Energy Harvesting. Journal on Communications, Vol. 39, 2018, No. 9, pp. 10–19, doi: 10.11959/j.issn.1000-436x.2018149.

[23] Tanyildizi, E.—Demir, G.: Golden Sine Algorithm: A Novel Math-Inspired Al-

gorithm. Advances in Electrical and Computer Engineering, Vol. 17, 2017, No. 2, pp. 71–78, doi: 10.4316/aece.2017.02010.

[24] Solis, F. J.—Wets, R. J. B.: Minimization by Random Search Techniques. Mathematics of Operations Research, Vol. 6, 1981, No. 1, pp. 19–30, doi: 10.2307/3689263.

[25] Heidari, A. A.—Mirjalili, S.—Faris, H.—Aljarah, I.—Mafarja, M.—Chen, H.: Harris Hawks Optimization: Algorithm and Applications. Future Generation Computer Systems, Vol. 97, 2019, pp. 849–872, doi: 10.1016/j.future.2019.02.028.

Yao Chen received her M.Sc. from the Xi'an Polytechnic University, in 2017. She is currently a Lecturer at Xi'an Key Laboratory of Human-Machine Integration and Control Technology for Intelligent Rehabilitation, School of Computer Science, Xijing University, China. Her research interests include optimization algorithms and applications.

Si Chen received her Ph.D. degree in mathematics from the School of Mathematics and Statistics, Northwestern Polytechnical University, China, in 2022. She is currently a Lecturer at the School of Science, Xi'an Shiyou University. Her research interests include optimization algorithms and applications.

Jian-Hui Li received his M.Sc. from the Xi'an Polytechnic University in 2013. He is currently serving as Professor at the Xi'an Key Laboratory of Human-Machine Integration and Control Technology for Intelligent Rehabilitation, School of Computer Science, Xijing University, China. His research interests include probability statistics.