



Boosting Whale Optimizer with Quasi-Oppositional Learning and Gaussian Barebone for Feature Selection and COVID-19 Image Segmentation

Jie Xing¹ · Hanli Zhao¹ · Huiling Chen¹ · Ruoxi Deng¹ · Lei Xiao¹

Received: 11 August 2022 / Revised: 9 October 2022 / Accepted: 19 October 2022 / Published online: 28 November 2022
© Jilin University 2022

Abstract

Whale optimization algorithm (WOA) tends to fall into the local optimum and fails to converge quickly in solving complex problems. To address the shortcomings, an improved WOA (QGBWOA) is proposed in this work. First, quasi-opposition-based learning is introduced to enhance the ability of WOA to search for optimal solutions. Second, a Gaussian barebone mechanism is embedded to promote diversity and expand the scope of the solution space in WOA. To verify the advantages of QGBWOA, comparison experiments between QGBWOA and its comparison peers were carried out on CEC 2014 with dimensions 10, 30, 50, and 100 and on CEC 2020 test with dimension 30. Furthermore, the performance results were tested using Wilcoxon signed-rank (WS), Friedman test, and post hoc statistical tests for statistical analysis. Convergence accuracy and speed are remarkably improved, as shown by experimental results. Finally, feature selection and multi-threshold image segmentation applications are demonstrated to validate the ability of QGBWOA to solve complex real-world problems. QGBWOA proves its superiority over compared algorithms in feature selection and multi-threshold image segmentation by performing several evaluation metrics.

Keywords Whale optimization algorithm · Quasi-opposition-based learning · Gaussian barebone · Image segmentation · Feature selection · Bionic algorithm

1 Introduction

Due to the rapid development of various industries, people face more complex optimization problems in real life. Conventional optimization techniques have limitations in resolving complex and massive optimization problems,

which cannot meet the requirement of convergence speed and calculation accuracy [1]. Compared with conventional optimization techniques, meta-heuristic algorithms (MAs) have the characteristics of flexibility, simplicity, derivation-free mechanism, and local optimal avoidance [2]. Therefore, MAs have been widely used to resolve complex optimization issues in recent years. The research inspiration for MAs mainly comes from biological behavior or natural physical phenomena. Furthermore, according to the different simulated natural behaviors, MAs can be divided into three main classes: evolution-based algorithms (EAs), physical-based algorithms (PAs), and swarm-based algorithms (SAs) [3]. EAs are inspired by Charles Darwin's theory of natural selection, in which the best individuals always combine to produce better offspring [4]. The main representatives of the evolution-based algorithm are genetic algorithms (GA) and differential evolution (DE). PAs imitate the physical rules and chemical reactions of the universe, such as simulated annealing (SA), particle swarm optimization (PSO), and sine cosine algorithm (SCA) [5]. SAs is a kind of heuristic algorithm that simulates the swarm behavior to solve

✉ Hanli Zhao
hanlizhao@wzu.edu.cn

✉ Huiling Chen
chenhuiling.jlu@gmail.com

✉ Lei Xiao
xiaolei@wzu.edu.cn

Jie Xing
xingjie095@163.com

Ruoxi Deng
ruoxii.deng@gmail.com

¹ Key Laboratory of Intelligent Informatics for Safety & Emergency of Zhejiang Province, Wenzhou University, Wenzhou 325035, China

optimization problems, such as firefly algorithms (FA) [6], moth flame optimization (MFO) [7], flame optimization algorithm (FOA) [8], Harris Hawks optimizer (HHO) [9], and Slime Mould algorithm (SMA) [10]. SAs have become an important method for solving optimization problems because of their excellent self-organization, self-adaptation, and self-learning characteristics. It has been adopted in various domains [11, 12], such as image segmentation [13], wireless networks [14], unmanned aerial vehicles [15], target tracking [16], neural network [17], MRI classification [18], feature selection [19], and engineering problems [20], and vehicle design [21, 22].

Mirjalili and Lewis proposed a novel meta-heuristic algorithm in 2016 [23], named whale optimization algorithm (WOA), which was encouraged by the humpback whale's foraging behavior. In recent years, WOA has attracted much attention and has been utilized to find optimal solutions in many fields. For example, based on chaotic and multi-swarm strategies, Wang et al. [24] developed an improved WOA and used it in two optimization scenarios. Chen et al. [25] introduced a chaotic local search strategy and Levy flight (LF) into WOA (BWOA), which was applied to solve three well-known problems in mathematical modeling studies. Abdel-Basset et al. [26] proposed two new variants of WOA, based on the ranking method and cyclic exploration–exploitation operator, and named RWOA and HWOA, respectively. They were applied to identify the parameters of the three-diode photovoltaic model. Peng et al. [27] proposed a hybrid WOA to improve the performance of cloud load forecasting. Chakraborty et al. [28] proposed a modified WOA variant and used it to solve problems in the engineering domain. Peng et al. [27] introduced a hybrid WOA based on the Levy and migration strategies into cloud load forecasting. Ye et al. [29] devised a novel modified WOA using the strategy of LF and pattern search and applied it to the field of energy optimization. Mostafa et al. [30] studied a WOA-based liver segmentation method from magnetic resonance images. Chao et al. [31] embedded orthogonal crossover into WOA to improve its exploration ability and applied it to estimate the surface duct. Hassib et al. [32] combined WOA with bidirectional recurrent neural network algorithms to train a deep learning approach. M.Hassi et al. [32] combined WOA with bidirectional recurrent neural network algorithms to train a deep learning approach. Darwish et al. [33] developed a novel whale optimization algorithm based on chaotic maps, which was used to select feature sets with high classification performance and a small number of features. Tripathi et al. [34] proposed a WOA variant for the recommendation over large-scale datasets in managing large-scale datasets.

In the original WOA, although the number of parameters to be adjusted is less and the convergence ability is strong

enough, it still has the disadvantages of slow convergence speed and low convergence accuracy. Therefore, many optimization schemes were proposed to mitigate them to overcome these shortcomings. Hussien et al. [35] introduced a binary whale optimization algorithm based on two transfer functions. Chakraborty [36] et al. improved WOA in three aspects: the original algorithm's parameters, the prey's search range, and the inertia weights. While reducing the complexity of the algorithm, it effectively improves the performance of the algorithm. Luo et al. [37] improved WOA based on the regularity of chaos and the mutational character of Gaussian mutation. Saha et al. [38] proposed a cosine-adapted modified whale optimization by incorporating cosine parameters into the selection of control parameters. In the study [1], LF and ranking-based mutation operators were embedded into WOA, which can prevent the algorithm from falling into local optimum and help the algorithm find the optimal solution quickly. Tu et al. [39] enhanced WOA with the strategies of communication mechanism and biogeography-based optimization algorithm, which overcome the shortcomings of WOA in slow convergence speed and easy trapping into the local optimum. Heidari et al. [40] integrated the learning mechanism and hill-climbing local search into WOA to enhance the exploitation process of the original WOA, which is called BMWOA. Chen et al. [41] combined two strategies, random replacement and double adaptive weight, with WOA to improve the performance of WOA. In the study [42], the exploration and exploitation capabilities of WOA were enhanced by embedding the modified mutualism phase of symbiotic organisms search and the DE mutation operator. Wang et al. [43] integrated elite strategy and spiral motion from MFO into WOA. Elhosseini et al. [44] introduced an inertia weight strategy to improve the parameters of WOA nonlinearly as a way to strengthen the ability of WOA to find optimal solutions. Chakraborty et al. [45] improved the exploration ability of WOA using the DE mutation strategy and balanced the exploration and exploitation capabilities by introducing a new parameter. Wang et al. [46] used opposition-based learning and a global grid ranking mechanism to enhance WOA's performance.

Although the above meta-heuristic algorithms, including WOA and improved WOA variants, have demonstrated their effectiveness in many optimization problems, there are still some shortcomings in the convergence speed and accuracy [11, 47]. The original WOA can be further improved when solving some complex practical problems, especially in the field of feature selection [48] and image segmentation [49]. In the late iterative stage, the exploitation capabilities of WOA do not enable much change in the location of search agents [44, 50]. When the algorithm is trapped in a local optimum, it does not solve this type of situation well.

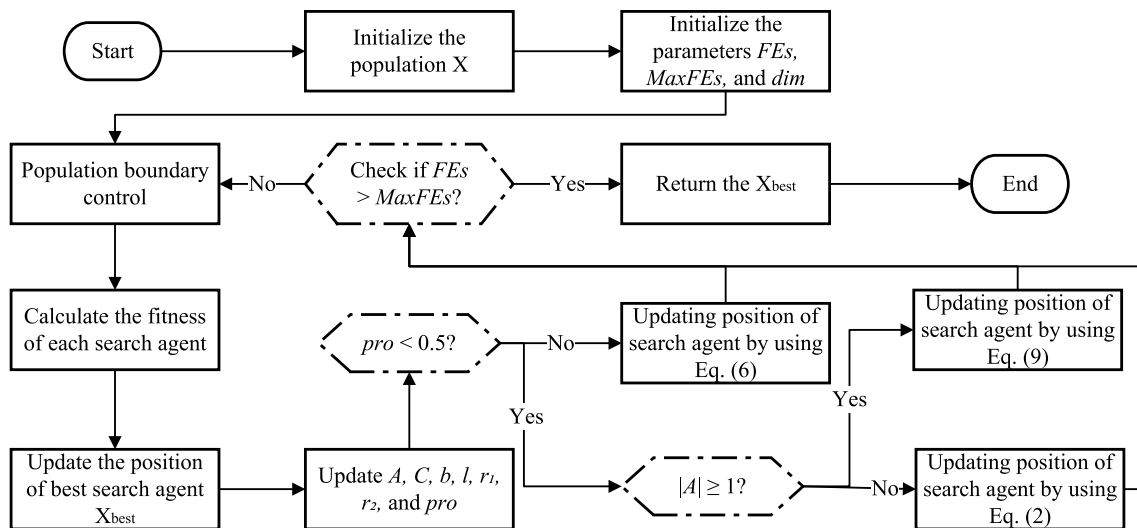


Fig. 1 The flow chart of WOA

This may make the WOA-based multi-threshold image segmentation, and feature selection methods have insufficient power to jump out of the local optimum and thus fail to achieve satisfactory optimization performance. In addition, the diversity of the population decreases with the number of evaluations [29, 41, 51]. The global exploration capability of WOA is insufficient, which may result in some regions with better solutions not being found. As a result, when WOA is applied in the multi-threshold image segmentation problem and feature selection problem, it may miss the thresholds and feature subsets that have better effects.

To this end, we designed an improved WOA, called QGBWOA, by combining a quasi-opposition-based learning (QOBL) strategy and the Gaussian barebone (GB) mechanism. To address the shortcoming of WOA in jumping out of the local optimum in the later evaluation, the QOBL strategy is introduced. QOBL strategy is characterized by considering the opposite position of the current solution in the solution space. Therefore, if the current solution falls into a local optimum, its opposite solution may help it to break out of the local optimum and find a more optimal solution. The local exploitation capability of the algorithm is thus enhanced. The GB mechanism is introduced to address the problem of population diversity decreasing in the late evaluations. GB mechanism can generate the position of individuals in the search region by Gaussian distribution. Therefore, it enhances the population's diversity and improves the algorithm's global exploration ability, which gives the algorithm more possibility to move to regions with better solutions for searching.

In summary, the main contributions of this paper are as follows:

1. A new enhanced WOA algorithm combining QOBL and GB, called QGBWOA, is proposed, with the experimental results showing that QGBWOA has higher accuracy and a faster convergence rate in obtaining global solutions.
2. A QGBWOA-based wrapper feature selection method is proposed for tackling feature selection tasks.
3. A QGBWOA-based image segmentation method of 2D histograms combined with Kapur's entropy is proposed and applied to real COVID-19 pathology images.
4. QGBWOA achieves higher classification accuracy and a smaller number of features in the feature selection task and shows excellent performance in multi-threshold image segmentation problems in all three evaluation metrics, including Peak Signal to Noise Ratio (PSNR) [52], Structural Similarity (SSIM) [53], and Feature Similarity (FSIM) [54].

The rest of this article is organized as follows. The original WOA is presented in Sect. 2. Section 3 describes the proposed QGBWOA algorithm. Section 4 analyzes the experimental results of QGBWOA in the benchmark function test. Section 5 provides the application of QGBWOA to the feature selection and image segmentation problems. The conclusion and future work are summarized in Sect. 6.

2 Whale Optimization Algorithm

WOA [23] simulates the hunting actions of whales: encircling prey, bubble net attack, and searching for prey.

2.1 Encircling Prey Phase (Exploitation)

In the exploitation phase, whales use bubble nets to attack their prey, including two models of shrinking encircling and spiral updating. In the encircling prey stage, the best search agent position obtained so far is selected as the optimal position, and other individuals gradually approach the best agent. Its mathematical model is as Eq. (2):

$$D_{\text{dist}} = |C \cdot X_{\text{best}}(t) - X(t)|, \quad (1)$$

$$X(t+1) = X_{\text{best}}(t) - A \cdot D_{\text{dist}}, \quad (2)$$

where $X(t+1)$ is the position of the search agent in the next evaluation, $X(t)$ is the position of the search agent in the current evaluation, and $X_{\text{best}}(t)$ is the best agent explored so far. Let FEs represent the counter of evaluation, A and C are two control parameters that can be presented as follows:

$$A = 2a \cdot r_1 - a, \quad (3)$$

$$a = 2 - 2 \times \frac{\text{FEs}}{\text{MaxFEs}}, \quad (4)$$

$$C = 2r_2, \quad (5)$$

where r_1, r_2 are two random numbers in [1], MaxFEs denotes the maximum number of evaluations of the algorithm, and a linearly decreases from 2 to 0 over evaluations.

2.2 Spiral Updating Phase (Exploitation)

The spiral updating phase is realized by Eq. (6). $O_{\text{dist}} = |X_{\text{best}}(t) - X(t)|$ denotes the distance between the search agent in the current evaluation and the best agent obtained so far.

$$X(t+1) = O_{\text{dist}} \cdot e^{fl} \cdot \cos(2\pi l) + X_{\text{best}}(t), \quad (6)$$

where f is a constant that controls the logarithmic spiral's shape and is set to 1 according to the original text. The parameter l is a random number in $[-1, 1]$.

The probability of being selected for the spiral moves and shrinking encircling phase is 50% each:

$$X(t+1) = \begin{cases} X_{\text{best}}(t) - A \cdot D_{\text{dist}} & \text{pro} < 0.5; \# \\ O_{\text{dist}} \cdot e^{fl} \cdot \cos(2\pi l) + X_{\text{best}}(t) & \text{pro} \geq 0.5, \# \end{cases} \quad (7)$$

where pro is a random number between 0 and 1.

2.3 Search for Prey Phase (Exploration)

When A is less than -1 or more than 1 , whales use a random walk mechanism to search for prey based on the locations of other individuals. The mathematical model for the exploration phase is as follows:

$$D_{\text{dist}} = |C \cdot X_{\text{rand}}(t) - X(t)|, \quad (8)$$

$$X(t+1) = X_{\text{rand}}(t) - A \cdot D_{\text{dist}}, \quad (9)$$

where $X_{\text{rand}}(t)$ represents a random search agent selected from the current population.

The flow chart of WOA is as shown in Fig. 1.

3 The proposed QGBWOA

In this section, the proposed QGBWOA will first be described by flowchart and pseudo-code. Then, the two strategies, QOBL and GB, will be described in detail. Finally, the time complexity of QGBWOA is analyzed.

3.1 Algorithm Overview

The flow chart of QGBWOA is shown in Fig. 2. We present the pseudo-code of the proposed QGBWOA in Algorithm 1.

The pipeline of QGBWOA is described as follows. First, the population is randomly initialized. Then, find the optimal individual in the current evaluation based on the fitness value, and update the individual position. In the position updating phase, by incorporating the QOBL mechanism, the ability of the algorithm to find a superior solution is boosted to some extent. Thus, the convergence rate and the quality of the solution can be improved. After the individual position is updated, the GB strategy is used to update the population position again. The increased diversity of the population enhances the exploration ability; thereby, the frequency of the algorithm falling into the local optimum is significantly reduced. The details of QOBL and GB mechanisms are presented in the following subsections.

Algorithm 1 Pseudo-code of QGBWOA

```

1. Initialization of initial parameters  $N$ ,  $dim$ ,  $FES$ ,  $MaxFES$ , and population  $X_i (i = 1, 2, 3 \dots N)$ 
2. While ( $FES < MaxFES$ ) do
3.   for  $i = 1: N$  do
4.     Calculate the fitness of  $X_i$ 
5.     if (the fitness of  $X_i < X_{best}$ ) then
6.       Update the position and the fitness of  $X_{best}$ 
7.        $FES = FES + 1$ 
8.     end if
9.   end for
10.  Update  $a$ ,  $a2$ 
11.  for  $i = 1: N$  do //update the position of each search agent
12.    Update  $A$ ,  $C$ ,  $b$ ,  $l$ , and  $pro$ 
13.    for  $j = 1: dim$  do
14.      if ( $pro < 0.5$ ) then
15.        if ( $|A| \geq 1$ ) then
16.          Update  $D_{dist}$ 
17.          Update  $X_{i,j}$  using Eq. (9)
18.        else
19.          Update  $X^o$  using Eq. (11)
20.          Update  $X_{i,j}^{qo}$  using Eq. (10)
21.        End if
22.      else
23.        Update  $O_{dist}$ 
24.        Update  $X_{i,j}$  using Eq. (6)
25.      End if
26.    End for
27.    if (the fitness of  $X_i^{qo} <$  the fitness of  $X_i$ ) then //perform the QOBL mechanism
28.      Update the position of  $X_i$ 
29.       $FES = FES + 2$ 
30.    End if
31.  End for
32.  for  $i = 1: N$  do //perform the GB mechanism
33.    Select three random search agents  $X_{t1}$ ,  $X_{t2}$ ,  $X_{t3}$ 
34.    Calculate  $V_i$  using Eq. (12)
35.    Calculate the fitness of  $V_i$ 
36.     $FES = FES + 1$ 
37.    Update  $X_i$  if there is a better solution
38.  End for
39. End while
40. Return the best solution.

```

3.2 Quasi-Opposition-Based Learning

As shown in Eq. (3), the value of parameter A will be less than 1 at the late stage of evaluation in the original WOA, and the encircling prey phase is executed. The position of

the new individual is only related to the position of the optimal individual and current individual, so the final new individual's position will not change significantly in the late evaluation, which may cause it to fall into the local optimum. Therefore, QOBL [55] is taken to enhance the original WOA in local search ability in the late evaluation process, to

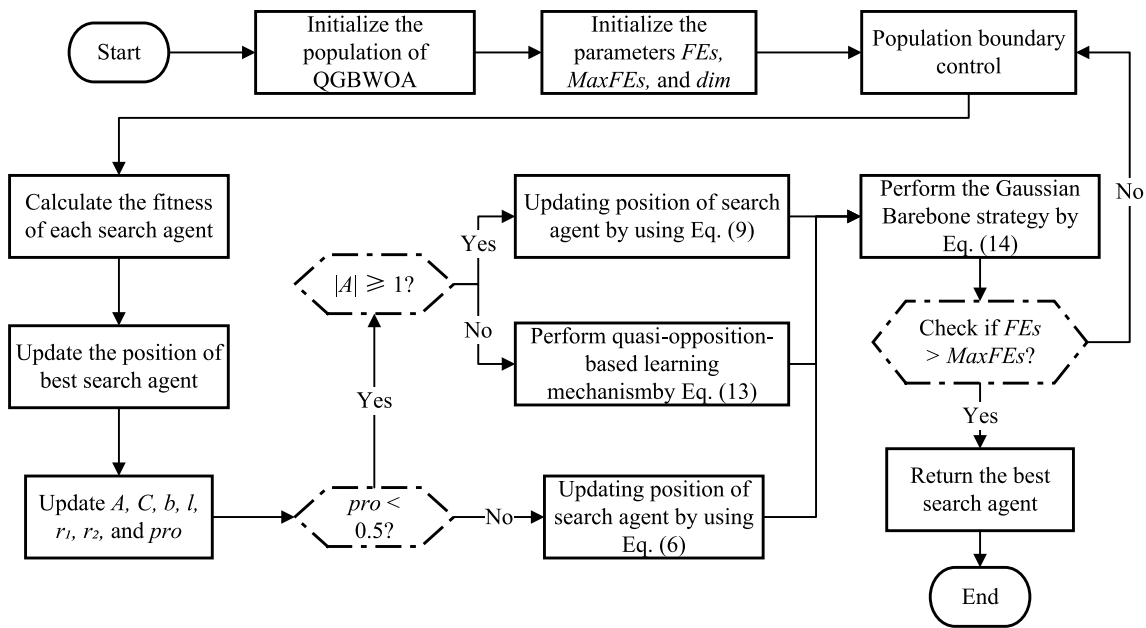


Fig. 2 Flow chart of QGBWOA

reduce the frequency of WOA falling into the local optimum. QOBL is an improved version of opposition-based learning (OBL) [56], which considers the individual with the opposite position to the current individual may be closer than the current individual. In recent years, the QOBL strategy has been used in MAs [57–60] to improve convergence speed accuracy.

The mathematical model is depicted as follows:

$$x_j^{qo} = rand\left[\left(\frac{lb_j + ub_j}{2}\right), x_j^o\right], \# \tag{10}$$

where x_j^{qo} represents the quasi-opposite individual of the current search agent in j -th dimension, $\frac{lb_j + ub_j}{2}$ represents the center of $[lb_j, ub_j]$, $rand\left[\left(\frac{lb_j + ub_j}{2}\right), x_j^o\right]$ represents a uniformly distributed random number between in $\frac{lb_j + ub_j}{2}$ and x_j^o , x_j^o denotes the opposite individual of the current search agent in j -th dimension:

$$x_j^o = lb_j + ub_j - x_j, \tag{11}$$

where lb_j denotes the lower bound of the search space, ub_j denotes the upper bound of the search space, and x_j denotes the position of the current individual in j -th dimension.

3.3 Gaussian Barebone Mechanism

As we mentioned earlier, when the algorithm is evaluated in the later phase, the diversity of WOA will reduce, which can cause insufficient convergence speed and convergence accuracy. The GB mechanism can help individuals choose

the most suitable direction and continuously approach the optimal solution to avoid prematurely falling into local optimality. Therefore, after the position of all the search agents has been updated, the characteristics of the randomness of GB are incorporated into WOA to enhance the population diversity. This balances the algorithm’s local exploitation and global search capabilities and further improves the convergence speed.

The GB [61] strategy is on the basis of bare bones PSO (BBPSO) [62], and the parameter CR is employed in the GB strategy to guide each individual. If the probability of random generation is less than CR, the Gaussian distribution is used to update the individual’s position in the next evaluation; otherwise, the idea of differential evolution is introduced to update the individual’s position. The GB strategy is as follows:

$$V_{i,j} = \begin{cases} G\left(\frac{P_{Leader} - X_{i,j}}{2}, |P_{Leader} - X_{i,j}|\right) & r_3 < CR; \\ X_{t1,j} + r_4 \times (X_{t2,j} - X_{t3,j}) & \text{otherwise,} \end{cases} \tag{12}$$

where $V_{i,j}$ denotes the position of the i -th individual in the j -th dimension, P_{Leader} denotes the global optimal position in the population, $X_{i,j}$ is the current individual in the j -th dimension, G represents the Gaussian distribution, r_3 and r_4 are random numbers within $[1]$, $X_{t1,j}$, $X_{t2,j}$, $X_{t3,j}$ are three arbitrarily selected individuals that are diverse from the current individual.

3.4 Time Complexity of QGBWOA

The time complexity of QGBWOA is subject to the population size (N), the number of dimensions (Dim), and the maximum number of algorithm evaluations (MaxFEs). Then, the overall time complexity is as follows:

- The population size is N . The time complexity of initializing all individual whales is $O(N)$.
- The time complexity of population fitness and updating the position and the fitness of the current optimal solution is $\text{MaxFEs} \times O(2N)$.
- The primitive WOA search mechanism also causes the position of each search agent to change during the search process of QGBWOA. The time complexity of updating the position of each search agent is $\text{MaxFEs} \times (O(N \times \text{Dim}) + 5 \times N)$.
- Implementing the QOBL mechanism is $\text{MaxFEs} \times O(N \times \text{Dim})$.
- Performing GB strategy is $\text{MaxFEs} \times O(N \times (\text{Dim} + 5))$.

Therefore, the total time complexity is $O(\text{QGBWOA}) = O(\text{Initialization}) + O(\text{Calculation of initial whales and Selection}) + O(\text{WOA}) + O(\text{QOBL strategy}) + O(\text{GB mechanism}) = O(N) + \text{MaxFEs} \times (O(2N) + O(N \times \text{Dim}) + 5 \times N + O(N \times \text{Dim}) + O(N \times (\text{Dim} + 5)))$.

4 Experimental Results and Discussion

In this section, the algorithm stability and strategy combination are analyzed, and experimental simulation results on the IEEE CEC 2014 and CEC 2020 benchmark functions are shown to verify the performance of QGBWOA comprehensively.

4.1 Benchmark Functions

CEC 2014 benchmark functions and CEC 2020 benchmark functions are used to verify the efficacy of QGBWOA. The details of the functions are shown in Appendix A.

For the fairness of the experiment results, all tested algorithms were performed in the same environment: the population size was 30, the maximum number of evaluations was set to 300,000, and the algorithms were independently

estimated 30 times on each benchmark function. We used the Friedman and WS tests to evaluate the experiment results.

4.2 Balance and Diversity Analysis on QGBWOA and WOA

In this section, QGBWOA is qualitatively analyzed on CEC 2014 in five aspects: search history, search trajectory, average fitness, and diversity and population balance.

The search history, search trajectory, and average fitness results are reported in Appendix B. In Fig. B.1, The first, second, third, and fourth columns of the figure show the three-dimensional of corresponding functions, the historical search trajectories in 2-dimensional (2D), the trajectories of the search agents, and the average fitness of individuals, respectively.

The red dots shown in Fig. B.1 (b) represent the global optimal solution, while the black dots represent the position of the search agent. The figure clearly shows that with the increase of the number of evaluations, the black dots gradually approach the red dots to find the optimal solution. In Fig. B.1 (c), the individual trajectory fluctuation is small in F2, while in the early evaluation process of F4, F10, and F16, the individual trajectory fluctuation is relatively strong. This shows that QGBWOA can reach most of the search space. Figure B.1 (d) shows that the average fitness decreases faster in F2, while the average fitness has strong fluctuations during early evaluations in F4 and F10. This indicates that QGBWOA can quickly determine the approximate range of optimal solutions in the early evaluations and further explore the optimal solution in later evaluations to achieve accurate convergence.

Figure B.2 shows the result of QGBWOA and WOA balance analysis. The figure's red, blue, and green curves represent the exploration, exploitation, and incremental-decremental curves. When the exploration effect is weaker than the exploitation effect, the green curve decreases, and vice versa. This is because the algorithm usually performs the global exploration first in the solution space, determines the approximate solution location with superior quality, and then performs the solution's local exploitation to find a more optimal solution. As shown in Fig. B.2, in the beginning of the evaluation, the search curve always starts with a higher value and the algorithm is mainly based on a global search. Then, local exploitation soon dominated. From Fig. B.2, it can be observed that the search phase of QGBWOA ends earlier than WOA, indicating that the local exploitation time of QGBWOA in the target area is longer.

Figure B.3 shows the results of QGBWOA and WOA diversity analysis. The diversity of the population is high at the beginning because the algorithm initializes the population randomly. As the evaluation progresses, the search range of the algorithm continues to decrease, and the population

Table 1 The ablation experiment of QGBWOA

	Q	GB
WOA	0	0
QWOA	1	0
GBWOA	0	1
QGBWOA	1	1

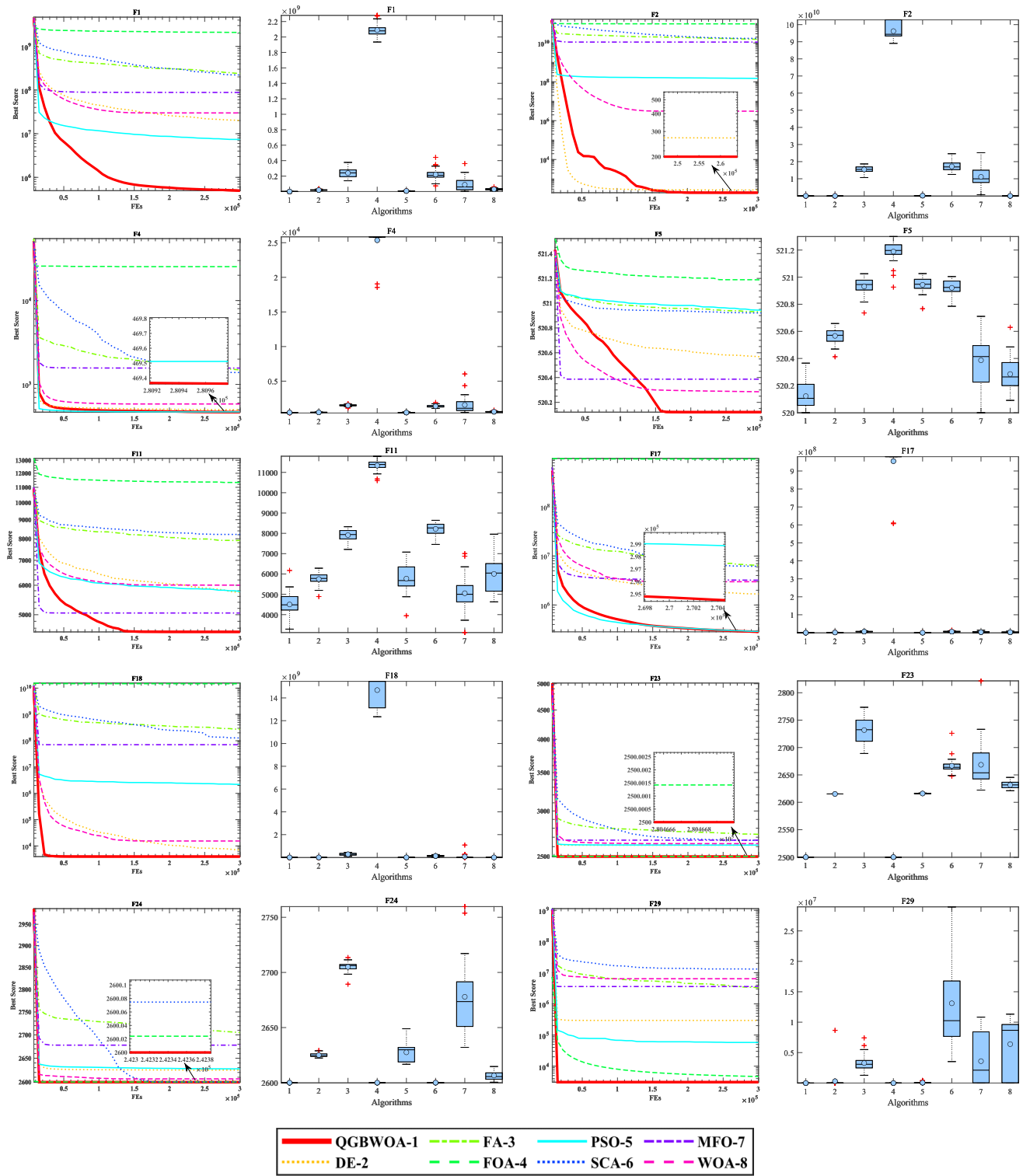


Fig. 3 Convergence curves and boxplots of QGBWOA and other meta-heuristics (Dim = 30)

diversity also continues to decrease. It can be seen from Fig. B.3 that the average diversity of QGBWOA falls faster than WOA, which indicates that QGBWOA converges more quickly than WOA.

In summary, QGBWOA demonstrates remarkable advantages in terms of convergence speed and global search capability compared with WOA.

4.3 Ablation Study on QGBWOA

To demonstrate the influence of QOBL and GB mechanisms, ablation experiments with QGBWOA were conducted. In Table 1, “Q” and “GB” mean the quasi-opposition-based learning mechanism and the Gaussian barebone mechanism, respectively. “1” implies that the corresponding mechanism is employed, and a “0” conversely indicates that it is not used. For example, QWOA indicates that WOA only uses the quasi-opposition-based learning mechanism and does not use the Gaussian barebone mechanism. Figure B.4 shows the convergence curves of QGBWOA with the other two mechanisms and the original WOA on the CEC 2014 benchmark functions, and the Dim is set to 30. It can be seen from the figure that QGBWOA is far superior to WOA in terms of convergence speed and convergence accuracy, which demonstrates the effectiveness of the GB mechanism in the global exploration capability of WOA. Experimental results show that QGBWOA is the best way to solve these different functions.

The WS test and Friedman test were used to compare the algorithms for statistical difference calculation. The results are shown in Appendix A. Table A. 3 records the average (Avg), standard deviation (Std), and average ranking (ARV) for each algorithm. “+”, “-” and “=” in Table A. 3 indicate that QGBWOA is better than other algorithms, inferior to other algorithms, and equal to other algorithms, respectively. Table A. 3 shows that QGBWOA ranks first among its two variants of the mechanism and the original WOA algorithm. On the benchmark functions F23–F25 and F27–F30, the Std values of QGBWOA are 0, which indicates that QGBWOA has good robustness. This is because the addition of the QOBL strategy improves the local exploitation ability of QGBWOA, and the addition of the GB strategy comprehensively improves the global exploration ability of QGBWOA, improves the population diversity, and better helps the algorithm find the global optimal solution. The final average ranking of QGBWOA is the first, which indicates that the algorithm is optimized best when these two mechanisms work together.

Table A. 4 shows the p values results of the WS test. When the value in the WS test is less than 0.05, QGBWOA has remarkable performance over its peer. The values less than 0.05 in the table have been bolded. From Table A. 3 shows that QGBWOA outperforms QWOA, GBWOA, and WOA in 20, 23, and 27 of the 30 benchmark functions, respectively. It can be seen that with the integration of the proposed two mechanisms, the performance is gradually improved.

4.4 Comparison with Other Metaheuristic Algorithms on CEC 2014 Test

In this section, QGBWOA is compared with 7 of the MAs, including WOA, DE [63], FA [6], FOA [64], PSO [65], SCA [5], and MFO [7]. Table A. 5 shows the results of F1–F30 with Dim values of 10, 30, 50, and 100, respectively.

It can be found that QGBWOA performs better in F1 when Dim values are 10, 30, 50, and 100. DE performs better on F2 and F3. Furthermore, compared with FOA, SCA, MFO, and WOA, QGBWOA is superior to them on all hybrid and composition functions when Dim is 10, 30, 50, and 100, respectively. DE can obtain the optimum on F15 in all four dimensions. QGBWOA finally ranks first when the Dim is 30, 50, and 100. The average mean is 1.8, which is 26% higher than the second-ranked DE algorithm and 63% higher than the original WOA. The p values are shown in Table A. 6 when Dim is 30. In Table A. 6 shows that the values of FA, FOA, and SCA on all functions are less than 0.05, indicating that QGBWOA outperforms these algorithms.

Figure 3 reports convergence curves and boxplots for each algorithm on 10 functions. In Fig. 3, F1 and F2 are unimodal functions. F4, F5, and F11 are multi-modal functions. F17 and F18 are hybrid functions. F23, F24, and F29 are composition functions. The first and third columns of the figure show the convergence curves, and the second and fourth columns show the corresponding box plots. In unimodal, multi-modal, and partial hybrid functions, although QGBWOA did not discover the optimal solution in the beginning phase, it converged to the optimal solution in later evaluations, indicating that QGBWOA has a good ability to avoid falling into the local optimal solution. In composition functions, the convergence rate and convergence accuracy of QGBWOA are significantly better than other algorithms. In the box plot, the center marker of each box indicates the intermediate value. Each box has the 25th and 75th percentiles at the lower and upper margins. Red “+” marks are used to mark outliers. The box plot in Fig. 3 shows that QGBWOA has more stable optimization results and fewer outliers than the compared algorithms in most cases. These results confirm the greatly improved performance of QGBWOA compared to WOA and other peers. The proposed QGBWOA has better performance due to the QOBL strategy and GB mechanism. The impact of these two strategies is shown in the convergence plots of the benchmark functions. Observing the convergence curves of F1, F2, F11, and F17, it can be found that the proposed algorithm does not fall into the local optimum at the evaluation times of about 50,000, and continues to converge to the region with higher solution quality, which illustrates that the QOBL strategy can be a good way to improve the local exploitation ability and the accuracy of the solution. In the convergence plots of F18, F23, F24, and F29, QGBWOA

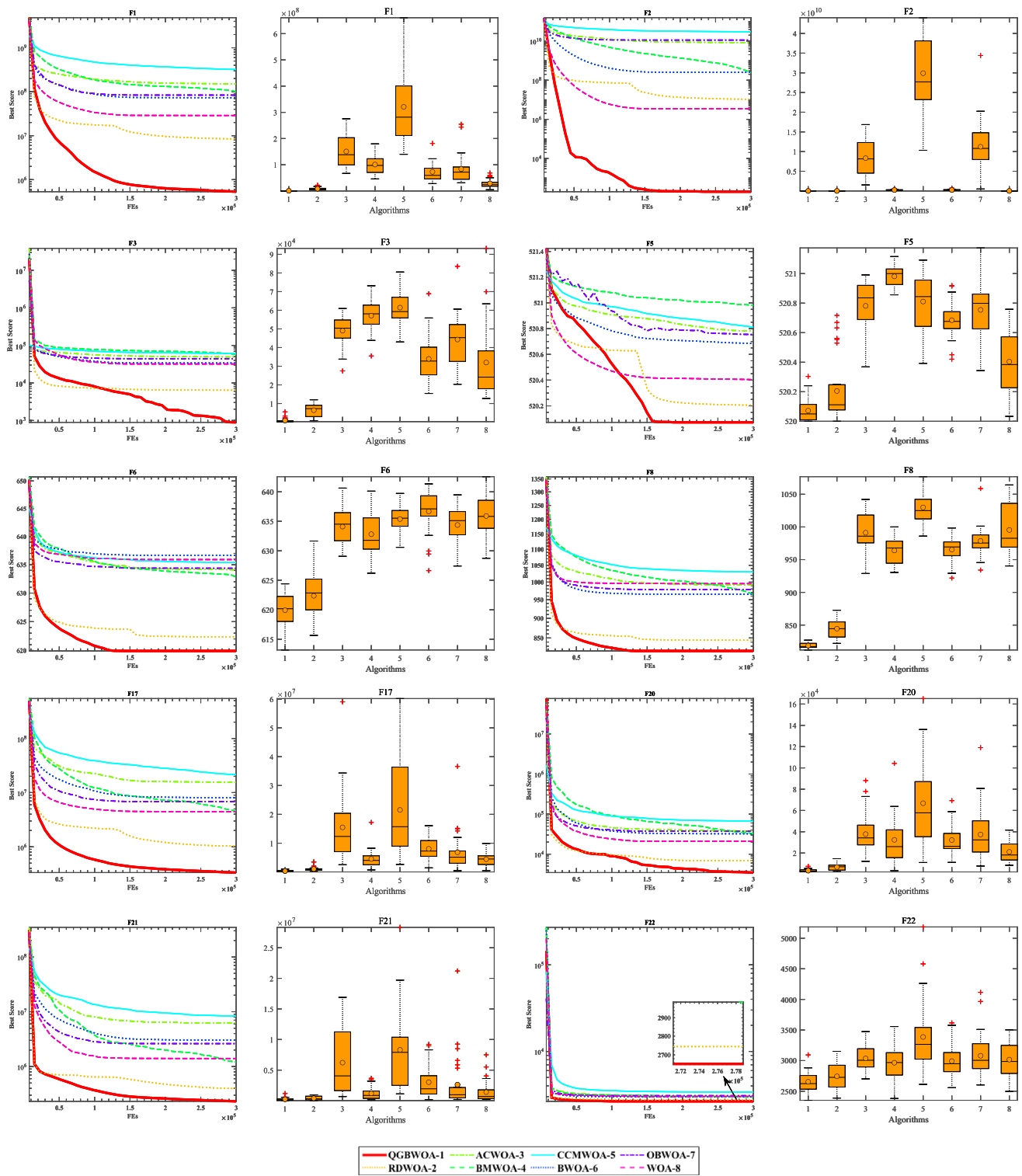


Fig. 4 Convergence curves and boxplots of QGBWOA and other variants algorithms

improves the exploration ability of the population due to the GB strategy so that the global optimal solution can be fast located, enabling the improvement of convergence faster.

4.5 Comparison with The State-of-The-Art WOA Variants on CEC 2014 Test

To confirm the efficacy of QGBWOA, we compared QGBWOA with 6 WOA variants on CEC 2014 test. These

Fig. 5 Bonferroni–Dunn test results of experiments in Sect. 4.4 (Dim = 30)

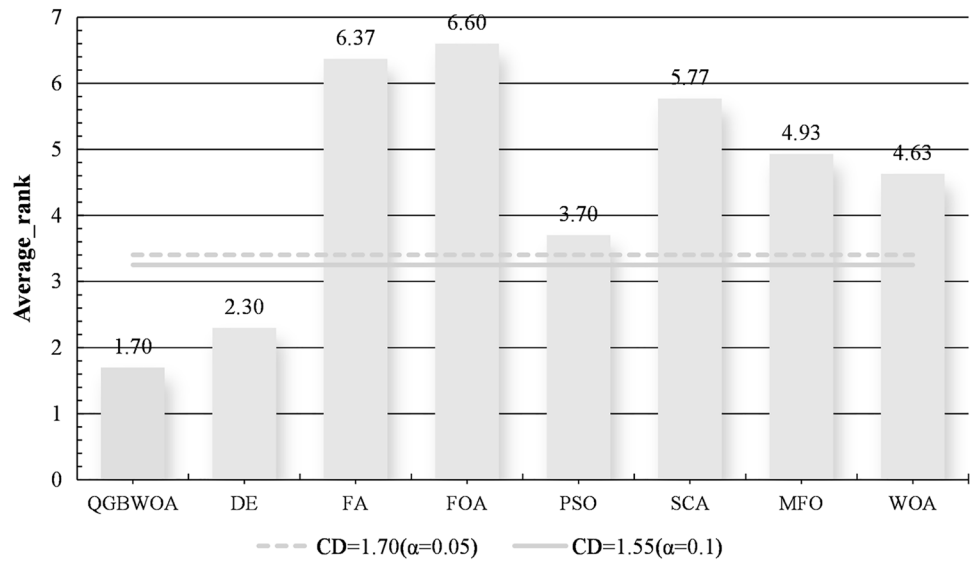
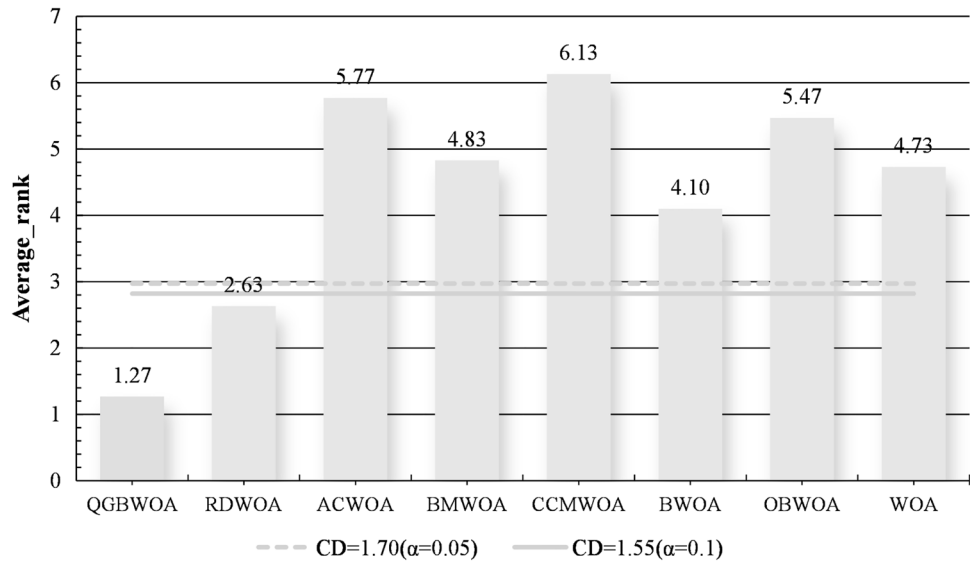


Fig. 6 Bonferroni–Dunn test results of experiments in Sect. 4.4.5 (Dim = 30)



variants include ACWOA [44], CCMWOA [37], OBWOA [46], RDWOA [41], BMWOA [40], and BWOA [25].

Table A.7 shows the statistical results in different dimensions. The table shows that QGBWOA obtains the smallest optimization result among the 30 test functions. QGBWOA ranks first when Dim values are 10, 30, 50, and 100, respectively. Compared with the second-ranked RDWOA, QGBWOA outperforms RDWOA in these four categories of dimensions with 8, 19, 19, and 21 functions, respectively. Moreover, QGBWOA performs better in all four dimensions on functions F1, F2, F4–F7, F15, F17, F19, F21–F23, and F27–F30.

Table A.8 reports the comparison results of the WS test between QGBWOA and the compared variants of WOA when Dim is 30. The *p* values on all test functions of

BMWOA except F14 are less than 0.05, which indicates that QGBWOA’s performance is superior to that of BMWOA. The *p* values of RDWOA, CCMWOA, and BWOA on the F23, F25, F27, and F28 functions are 1, indicating that RDWOA, CCMWOA, and BWOA can obtain the same optimization results as QGBWOA.

In Fig. 4, the convergence rate of QGBWOA is faster than the compared state-of-the-art algorithms on most of the benchmark functions. In addition, it can be observed that its convergence accuracy is the best, whereas other variants of WOA are trapped in local optimal values to varying degrees. Figure 4 depicts the box plots of the fitness of the best individuals found in the final generation. These comparison results show that QGBWOA is better than these state-of-the-art algorithms for complex optimization problems. The

results affirmed the ability of QGBWOA to solve benchmark problems in different dimensions.

4.6 Comparison of CEC 2020 Benchmark Functions

In this subsection, QGBWOA is tested with 7 group intelligence algorithms, including WOA, HHO [9], SMA [10], HGS [66], SCA [5], RDWOA [41], ACWOA [44], on the CEC 2020 benchmark functions with Dim equals to 30. Table A.9 illustrates the statistical results of all optimizers on Avg and Std. It can be seen in the table that QGBWOA outperforms the compared optimizers for 9, 8, 4, 3, 10, 6, and 8 of the 10 CEC 2020 benchmark functions, respectively. QGBWOA ranks first in total with ARV equal to 1.4.

The p value results of QGBWOA against the compared optimizers are shown in Table A.10, where p values greater than 0.05 are shown in bold. The results show that QGBWOA significantly differs from the optimizer for most of the tested functions except F9 and F10.

4.7 Statistical Analysis of QGBWOA

Because the Friedman test can only give a conclusion on whether there is a variance in performance among algorithms, a post hoc test is needed to find the statistical difference in the performance of the algorithms. Commonly used follow-up tests include the Nemenyi and Bonferroni–Dunn tests [67]. The Nemenyi test is employed to compare the performance of algorithms with each other, while the Bonferroni–Dunn test is used to compare an algorithm with the rest algorithms. Bonferroni–Dunn post hoc statistical analysis is used in this article to verify the performance difference between QGBWOA and the compared algorithms. Assume the difference in the average rank between the two algorithms is better than the critical difference (CD). The CD is described as Eq. (13).

$$CD = q_{\alpha} \sqrt{\frac{k(k+1)}{6Num}}, \quad (13)$$

where α denotes the significant level, q_{α} is the critical value, k is the number of algorithms, and Num represents the number of test functions.

In the experiment in Sect. 4.4, eight algorithms were chosen, so k is 8. Thirty benchmark functions were used, so $Num = 30$. The significant levels of α were selected as 0.05 and 0.1. According to Eq. (13), it can be calculated that when α values are 0.05 and 0.1, respectively, the corresponding CD values are 1.7 and 1.55, respectively. The average rank of QGBWOA is $ARV_{QGBWOA} = 1.70$ when Dim is 30. When the average rank of the comparison algorithm is greater than $CD + ARV_{QGBWOA} = 3.4/3.25$ ($\alpha = 0.05/0.1$), there is a significant difference between QGBWOA and this algorithm. In Fig. 5, the solid line represents the threshold

when the significant level is 0.1, and the dotted line indicates the threshold when the significant level is 0.05. As shown in Fig. 5, QGBWOA outperforms FA, FOA, PSO, SCA, MFO, and WOA at these two significant levels. In the Bonferroni–Dunn test, QGBWOA and DE showed no remarkable difference in performance.

The post hoc test analysis is performed for the experiment in Sect. 4.5. The results are shown in Fig. 6. Because eight algorithms were tested in 30 functions, the k is 8 and Num is 30. Similarly, CD values are 1.70 and 1.55 when the corresponding α values are 0.05 and 0.1, respectively. From the figure, we can see that QGBWOA is significantly better than ACWOA, BMWOA, CCMWOA, BWOA, OBWOA, and WOA at both significant levels and exhibits no obvious difference from RDWOA in the Bonferroni–Dunn test.

5 Applications

In this section, we will show the applications of QGBWOA in feature selection and image segmentation.

5.1 Feature Selection Based on Proposed QGBWOA

5.1.1 Binary QGBWOA

Data mining technology has been widely used in medicine in recent years. Most medical data are high-dimensional, and extracting a sub-set of useful features from high-dimensional data is tricky. Therefore, the dimensionality of the dataset needs to be reduced before data mining. Feature selection is one of the dimensionality reduction methods. However, it is often difficult to exhaust the combination of feature subsets, especially when dealing with high-dimensional data, and using a heuristic algorithm can solve this problem well [68]. As a heuristic method, the SAs have the intelligent selection and random search advantages, which can search for an ideal solution set. In recent years, the combination of SAs and feature selection methods has received more and more attention from researchers [69–72]. In this subsection, a QGBWOA-based wrapper feature selection method is proposed. Twenty-four datasets from the UCI machine learning repository are utilized to measure the effectiveness of the proposed method.

Since the search space in the feature selection problem is represented in binary, the real value of each solution found by QGBWOA must be converted into a Boolean type. The function is defined as follows:

$$X_{ij}^{t+1} \begin{cases} 1, & \text{if } r_5 \geq T(X_{ij}^t); \\ 0, & \text{otherwise,} \end{cases} \# \quad (14)$$

$$T(x) = \frac{1}{1+e^{-2x}}, \# \quad (15)$$

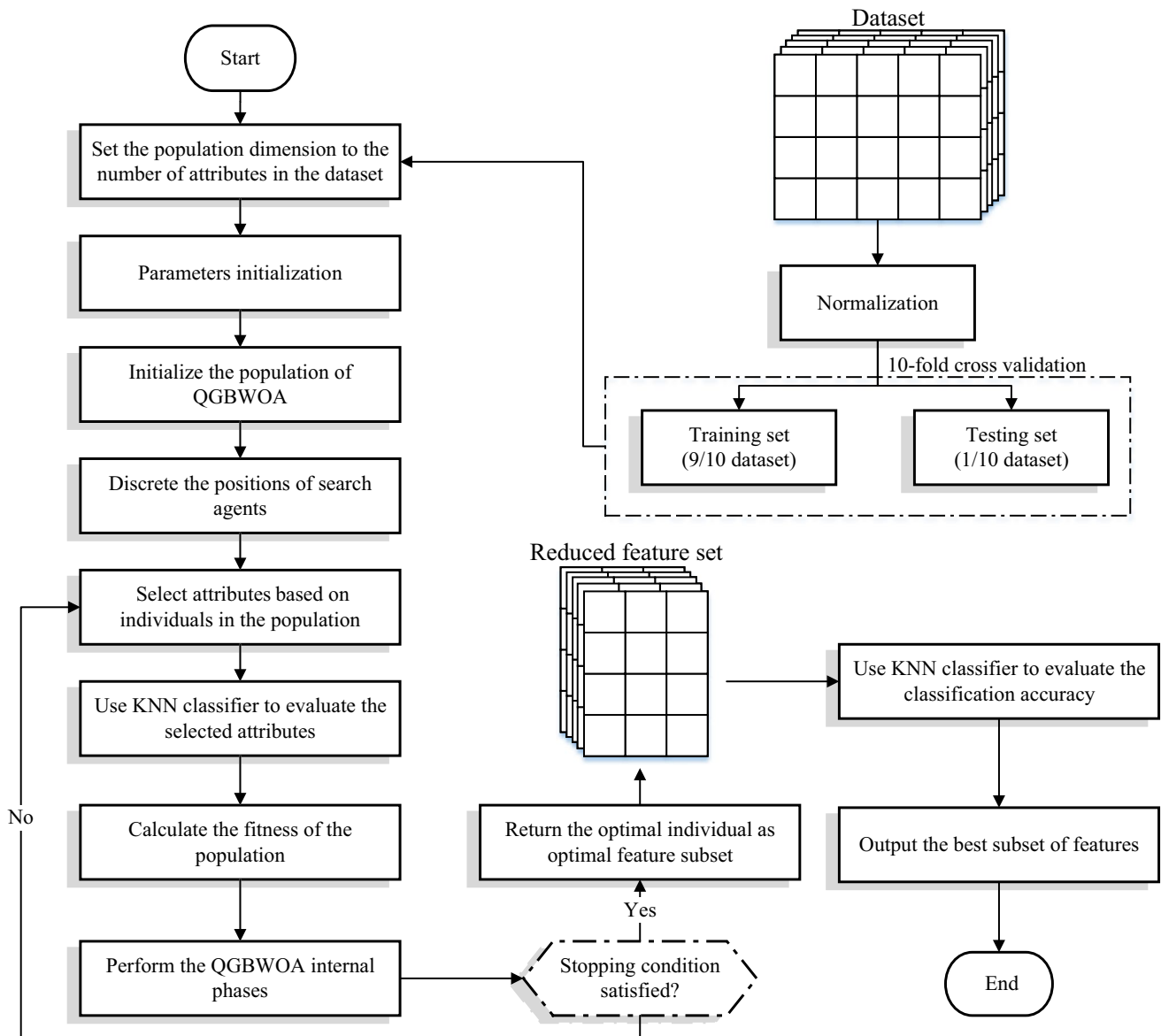


Fig. 7 The architecture of feature selection using the binary QGBWOA

where r_5 is a random number between in 0 and 1. If $X_{i,j} = 1$, it means the solution of X_i in the j -th dimension is regarded as a relevant feature; otherwise, if $X_{i,j} = 0$, it means the solution of X_i in the j -th dimension is regarded as an irrelevant feature. Since feature selection aims to obtain better classification accuracy with fewer features, the classifier error rate and the number of selected features are utilized to form the fitness function. The fitness function is defined as follows:

$$Fitness = \theta \cdot Acc + v \cdot \frac{S}{T}, \# \tag{16}$$

where Acc denotes the classification accuracy of the K -nearest neighbors (KNN) classifier, θ and v represent the weight coefficients of the classifier error rate and the number of

selected features, respectively, S represents the number of the selected feature sub-set, and T means the total number of features in the dataset.

The overall algorithm pipeline is shown in Fig. 7. First, the dataset is processed. Second, using ten-fold cross-validation to divide the dataset into ten parts, nine of which are used as training data, and the remaining one is used as test data. Before utilizing QGBWOA to search for the best feature combination in the dataset, the number of attributes of the dataset is set to the dimension of the population in QGBWOA. The KNN classifier is used to evaluate the accuracy of the selected features. The fitness of the population is calculated. Then, QGBWOA is applied to update the position of the population in the discrete search space. After reaching

Table 2 Datasets of the feature selection experiment

Category	Datasets	Samples	Attributes	Classes	
Low-dimension	Blood	997	69	2	
	Breast cancer	286	9	2	
	Congress	435	17	2	
	Ionosphere	351	35	2	
	Krvskp	208	61	2	
	M-of-n	100	14	2	
	Parkinson	195	23	2	
	Sonar	208	61	2	
	Spect	267	23	2	
	Heart	270	14	5	
	Lymphography	148	19	4	
	Penglung	73	326	7	
	High-dimension	Brain_tumor1	90	5921	5
		Brain_tumor2	50	10,368	4
CNS		60	7130	2	
DLBCL		77	5470	2	
Leukemia		72	7131	2	
Leukemia1		72	5328	3	
Leukemia2		72	11,226	3	
Lung_cancer		203	12,601	3	
Prostate_tumor		102	10,510	2	
SRBCT		83	2309	4	
Tumors_9		60	5727	9	
Tumors_11	174	12,534	11		

a specified number of evaluations, an optimal feature sub-set is obtained. The KNN classifier evaluates the classification accuracy of the obtained feature sub-set. Finally, the best sub-set of features is obtained.

5.1.2 Experiment on Feature Selection

Twelve low- and high-dimensional datasets are used to examine the efficacy of the proposed approaches. Both these two categories of datasets are chosen from the California

Irvine (UCI) Machine Learning Repository [73]. Table 2 describes datasets in terms of the number of samples, features, and classes. It can be seen that datasets in low-dimension contained less than 350 features, and most of the high-dimensional datasets have more than 5000 features. And QGBWOA was tested together with bMFO [74], BSMA [75], BSO [76], bWOA [77], and BDE [78]. Table 3 reports their parameter settings. Moreover, the datasets are divided by ten-fold cross-validation [79]. The wrapped feature selection method is based on the KNN ($K = 1$) classifier [80]. The maximum iteration is set to 50.

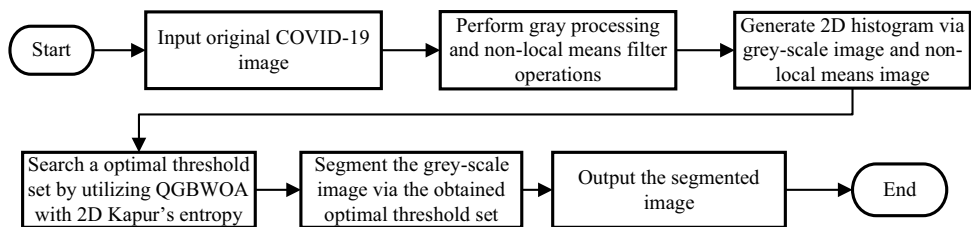
Appendix A reports the results of QGBWOA and other methods based on average classification error, the number of selected features, fitness values, and time cost. Table A.11–Table A.13 and Table A.15–Table A.17 show the fitness, number of selected features, and average classification error results on low-dimensional and high-dimensional datasets, respectively. It can be observed that QGBWOA significantly outperforms the compared algorithms in terms of fitness and can select a smaller number of features. QGBWOA ranked first in total compared to other methods. Table A.14 and Table A.18 show the timing results on each data set. The results in the table report that the time cost of the proposed QGBWOA is higher than the compared algorithms, such as bSMA and BSO. To a certain extent, the embeddedness of QOBL and GB strategies increases the time overhead of the proposed algorithm.

Figure B. 5 shows the convergence curves of QGBWOA and the other five algorithms on 12 low-dimensional datasets. The figure shows that QGBWOA performs well on all these 12 datasets. The convergence value of QGBWOA reaches the minimum, indicating that QGBWOA can obtain higher classification accuracy than the algorithms. Furthermore, it can be seen from Fig. B. 6 that the convergence value of the fitness of QGBWOA on all 12 high-dimensional datasets is smaller than that of the compared methods except Brain_tumor2 and Lung_cancer.

Table 3 Parameter setting of compared algorithms

Algorithm	Other parameters	Common parameters
BMFO	$b = 1$	Dim = number of features in the current data set; population size = 20
BSMA	$a =$	
BSO	$C1 = 0.5$	
	$C2 = 2$	
	Threshold = 0.25	
bWOA	$b = 1$ $r1 =$ $r2 =$	
BDE	beta_min = 0.2 beta_max = 0.8 pCR = 0.2	

Fig. 8 Flow chart of multi-threshold image segmentation using QGBWOA



5.2 Image Segmentation Based on Proposed QGBWOA

5.2.1 Proposed Image Segmentation Method

Image segmentation is a fundamental technique in a variety of image processing applications. The characteristics of the image divide the image into multiple discrete regions, which are characterized by continuity or similarity in the same region and have obvious contrast between different regions [81]. The multi-threshold method is a commonly used image segmentation method, which uses multiple thresholds to mark target regions of interest in an image. The choice of threshold has an impact on the image segmentation effect. One of the often-considered methods is the histogram-based

method. The histogram describes the frequency of the corresponding gray value in the image. The one-dimensional histogram only reflects the magnitude of the pixel's gray level, and the 2D histogram can reflect the spatial correlation information between the pixel and its neighborhood. Abutaleb et al. [82] proposed an image segmentation method combining the local pixel gray average with the original gray histogram. Kapur's entropy can be used to evaluate the optimal thresholds. It divides the image into different categories and determines whether the categories are consistent according to the entropy size. Kapur's entropy finds the optimal threshold by maximum fitness value [83]. Kapur's entropy is chosen in this article to determine the n best thresholds. Let $[T_1, T_2, T_3, \dots, T_n]$ be the threshold set values to divide the image into $n + 1$ classes and the formula is described as:

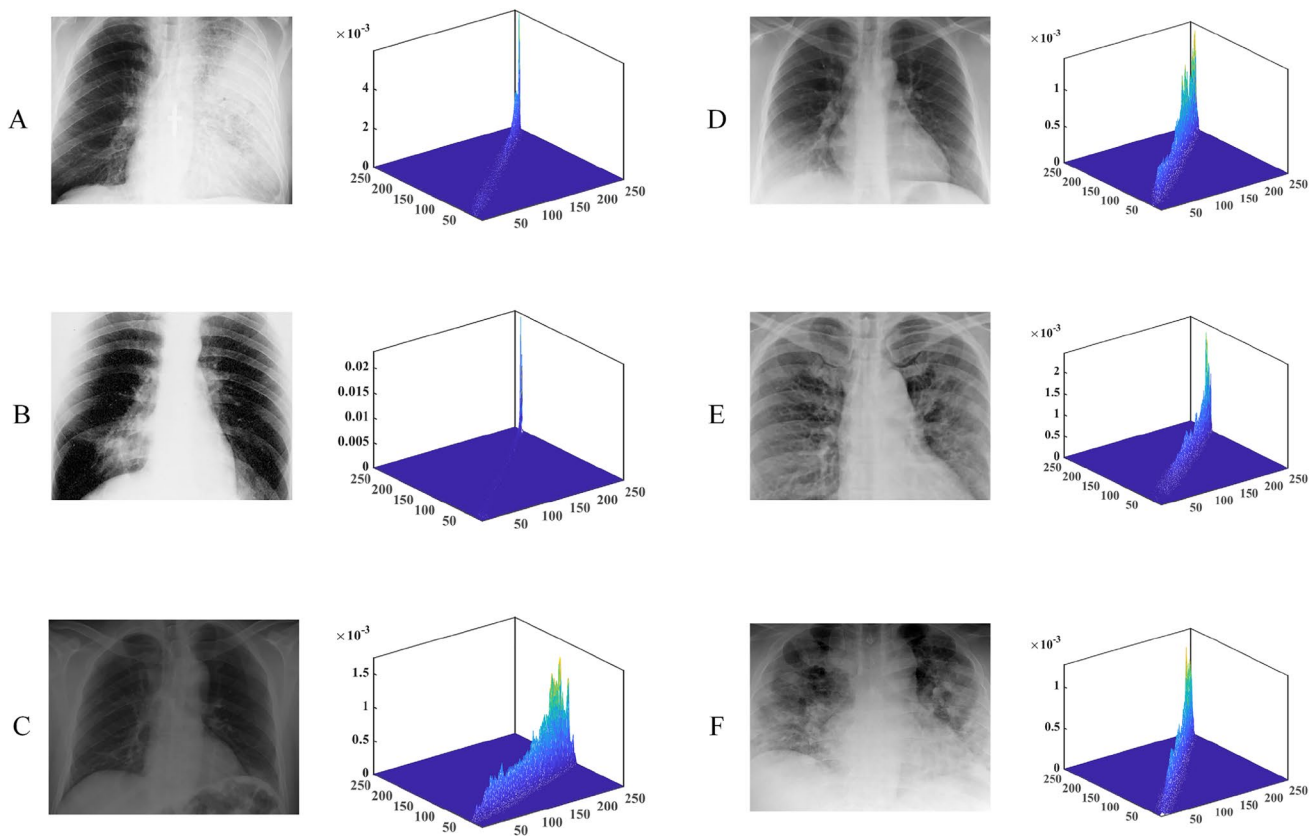


Fig. 9 The original images and their 2D histograms

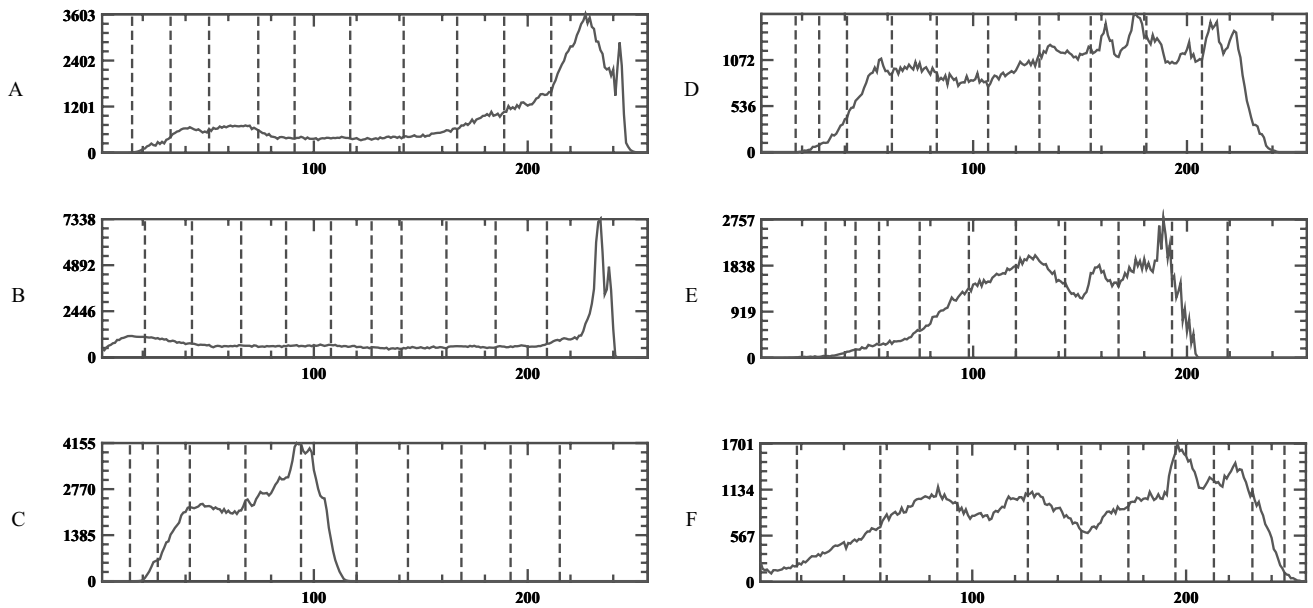


Fig. 10 The optimal threshold of images at threshold level 10

Table 4 PSNR average results rankings

Threshold	QGBWOA	WOA	CS	MFO	SSA	BLPSO
4						
ARV	3.500	4.000	3.833	3.000	3.000	3.667
Rank	3	6	5	1	1	4
6						
ARV	2.833	3.833	3.333	3.833	4.333	2.833
Rank	1	4	3	4	6	1
10						
ARV	3.167	3.500	4.167	2.833	3.333	4.000
Rank	2	4	6	1	3	5
15						
ARV	1.833	4.167	4.833	4.000	3.667	2.500
Rank	1	5	6	4	3	2

$$F(T_1, T_2, T_3, \dots, T_n) = H_0 + H_1 + \dots + H_n, \# \tag{17}$$

$$H_n = -\sum_{s=T_n}^{L-1} \frac{p_s}{\omega_n} \ln \frac{p_s}{\omega_n}, \omega_n = \sum_{s=T_n}^{L-1} p_s, \# \tag{20}$$

$$H_0 = -\sum_{s=0}^{T_1-1} \frac{p_s}{\omega_0} \ln \frac{p_s}{\omega_0}, \omega_0 = \sum_{s=0}^{T_1-1} p_s, \# \tag{18}$$

$$p_s = \frac{h(s)}{P}, s = 0, 1, \dots, L-1, \# \tag{21}$$

$$H_1 = -\sum_{s=T_1}^{T_2-1} \frac{p_s}{\omega_1} \ln \frac{p_s}{\omega_1}, \omega_1 = \sum_{s=T_1}^{T_2-1} p_s, \# \tag{19}$$

∴ where P is the total number of pixels in the image, L represents the total gray levels of the given image, s is the gray level, p_s is the probability of the s -th gray level, H_1, H_2, \dots, H_n denote the Kapur's entropies of corresponding classes, and $\omega_0, \omega_1, \dots, \omega_n$ denote the probabilities of corresponding classes.

Table 5 SSIM average results ranking

Threshold	QGBWOA	WOA	CS	MFO	SSA	BLPSO
4						
ARV	2.833	4.667	3.833	2.833	3.167	3.667
Rank	1	6	5	1	3	4
6						
ARV	2.500	2.500	3.833	3.833	4.500	3.833
Rank	1	1	3	3	6	3
10						
ARV	3.333	3.833	3.333	3.667	3.833	3.000
Rank	2	5	2	4	5	1
15						
ARV	2.667	4.167	3.667	4.333	2.833	3.333
Rank	1	5	4	6	2	3

Table 6 FSIM average results ranking

Threshold	QGBWOA	WOA	CS	MFO	SSA	BLPSO
4						
ARV	2.833	4.167	4.167	3	3	3.834
Rank	1	5	5	2	2	4
6						
ARV	2.667	3.667	4.000	3.500	4.167	3.000
Rank	1	4	5	3	6	2
10						
ARV	2.500	3.167	4.000	3.667	3.667	4.000
Rank	1	2	5	3	3	5
15						
ARV	2.500	4.333	4.167	4.500	2.500	3.000
Rank	1	5	4	6	1	3

Enumerating all combinations of thresholds and selecting the optimal one is quite difficult, and the time complexity will grow exponentially with an increasing number of thresholds [84]. The use of MAs to find the optimal threshold has attracted attention in recent years [85–88]. An improved butterfly optimization algorithm was proposed by Sharma et al. [89], which is utilized for image segmentation problems. Chakraborty et al. [90] introduced an enhanced version of WOA to tackle image segmentation problems. A new pulse couple neural network model based on grey wolf optimizer was proposed by Wang et al. [88] for medical image segmentation. Zhao et al. [91] improved the Salp swarm algorithm (SSA). This version is used to find optimal segmentation threshold for images.

In this subsection, we put forward a QGBWOA-based multi-threshold image segmentation method by integrating

the 2D histogram with the entropy of Kapur. QGBWOA is utilized to find the optimal set of thresholds, where the entropy of Kapur is the objective function, and the image is segmented according to the threshold. The detailed flowchart of the method is given in Fig. 8.

5.2.2 Simulation Experiment

We selected six COVID-19 patients’ images collected by Cohen et al. [92] as the segmentation images. These six images are named A, B, C, D, E, and F in this experiment. Computed tomography (CT) of the lungs of COVID-19 patients often shows high-gray diffuse ground-glass and pulmonary nodular shadows. This is because the COVID-19 virus enters the pulmonary bronchi and further invades the alveolar epithelial cells, where it replicates itself. The rapid

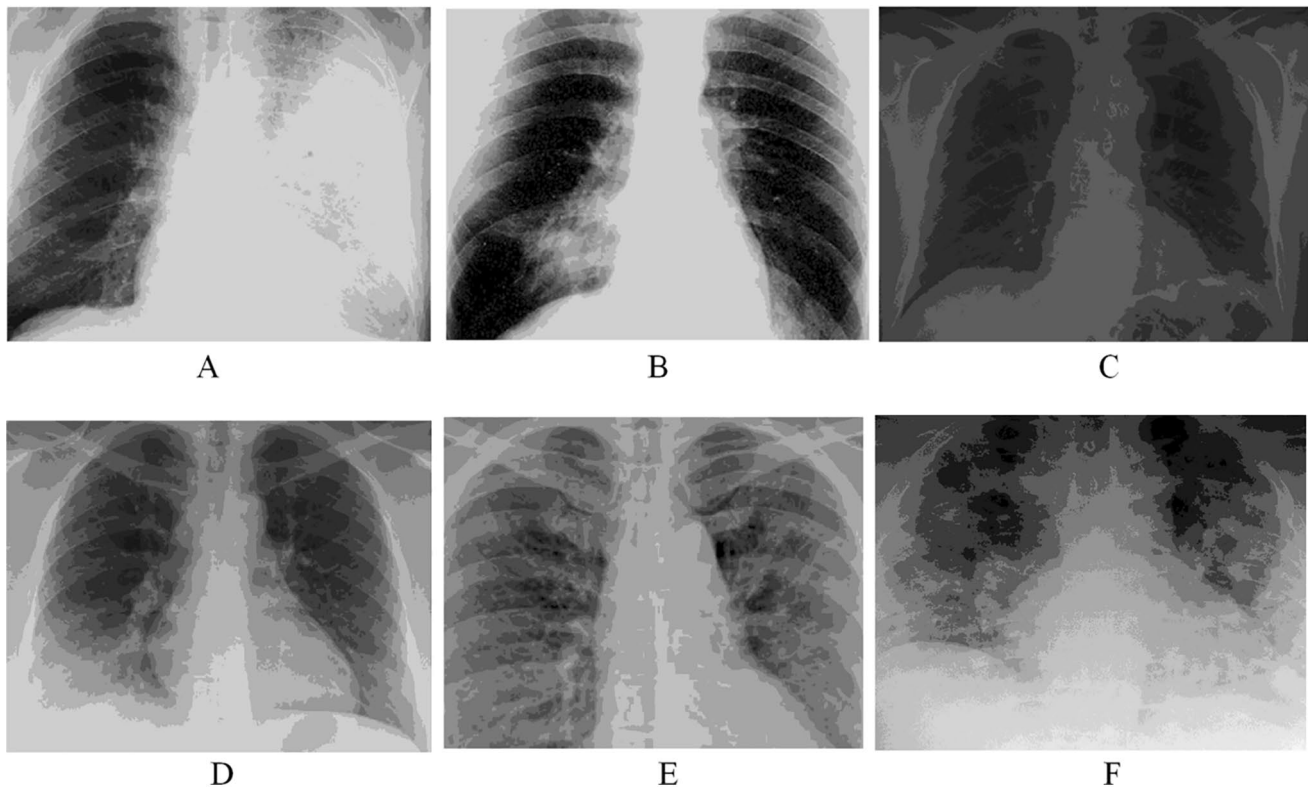


Fig. 11 The segmentation results of all images obtained by QGBWOA at threshold level 10

replication of the virus leads to significant swelling of the epithelial cells, which will appear as high-gray shadows on CT images with significant gray differences from the normal lung parenchyma.

Figure 9 shows the original images and the 2D histograms of these six images. The population number of the algorithm in this experiment was set to 20, the iteration times were set as 100, and the size of the image was set as 512×400 . The performance of the QGBWOA-based multi-threshold image segmentation method was evaluated at various thresholds, including low threshold levels (4 and 6) and high threshold levels (10 and 15). We compared our QGBWOA-based multi-threshold image segmentation with the multi-threshold segmentation methods based on other methods, including WOA, Cuckoo search (CS) algorithm, MFO, SSA, and biogeography-based learning PSO (BLPSO). In addition, the experiment results are evaluated by three indicators, including PSNR, SSIM, and FSIM.

The set of ideal segmentation thresholds found by the proposed algorithm for these images is shown in Fig. 10. The solid line in Fig. 10 represents the grayscale histogram, and the dotted line represents the ideal threshold set produced by QGBWOA. Because the threshold is 10, there are ten red lines in each picture.

The Avg and Std values of PSNR, SSIM, and FSIM in each threshold are shown in Appendix A, where the optimal results have been bolded. The results in Tables A.19 to Table A.21 show that QGBWOA achieves optimal results at most thresholds. The mean value of the overall ranking of these evaluation metrics is shown in Tables 4, 5 and 6. The results show that the mean of the overall ranking of QGBWOA is the smallest, which confirms the strong competitiveness of the proposed method. The best results for Kapur's entropy obtained by the proposed method are illustrated in Appendix A. In Table A.22, QGBWOA has a noticeable superiority in searching for the optimal value of Kapur's entropy compared with other algorithms. Figure 11 illustrates the final segmentation results with a threshold of 10. From the result, we can see that depending on the threshold found by QGBWOA; the image can be segmented into blocks of pixels with different gray values with sharper borders, which is useful for evaluating suspected cases of COVID-19.

6 Conclusion and Future Work

In this paper, a new WOA variant QGBWOA based on QOBL and GB strategy has been proposed to improve the inadequacies of the original WOA. The QOBL strategy is

introduced to strengthen the local exploitation ability and to assist the proposed method to jump out of the local optimum. The GB strategy is employed to balance the algorithm's exploitation and exploration capabilities and help the algorithm find regions with better solutions. QGBWOA was tested on CEC 2014 and CEC 2020 benchmark functions with different dimensions, in which the performance of QGBWOA is compared with the basic methods and the state-of-the-art WOA variants. The experimental results show that QGBWOA can provide optimum solutions and effectively avoid premature convergence. Finally, the ability of QGBWOA to solve real-world problems is validated by the feature selection and the multi-threshold image segmentation applications.

The increase in time complexity is the inherent result of mechanism embedding, and how to reduce the time complexity is also our next work. In future, parallel computing techniques [93] can be considered to reduce the time complexity while keeping the performance of QGBWOA. Additionally, we would like to apply QGBWOA to other fields, such as multi-objective optimization [94], fuzzy definition optimization [95], and dynamic optimization [96].

7 Data Availability Statement

The data involved in this study are all public data, which can be downloaded through public channels.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s42235-022-00297-8>.

Acknowledgements The authors would like to thank the editors and the anonymous reviewers for their valuable comments on improving this paper. Many thanks also to Ali Asghar Heidari for his helpful proofreading. This work was supported by the Zhejiang Provincial Natural Science Foundation of China (no. LZ21F020001) and the Basic Scientific Research Program of Wenzhou (no. S20220018).

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Yan, Z. P., Zhang, J. Z., Zeng, J., & Tang, J. L. (2021). Nature-inspired approach: An enhanced whale optimization algorithm for global optimization. *Mathematics and Computers in Simulation*, *185*, 17–46. <https://doi.org/10.1016/j.matcom.2020.12.008>
2. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, *69*, 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
3. Faris, H., Mirjalili, S., Aljarah, I., & Mafarja, M. (2020). *Nature-inspired optimizers: Theories, literature reviews and applications* (pp. 185–199). Springer International Publishing.
4. Li, C. Y., Li, J., Chen, H. L., Jin, M., & Ren, H. (2021). Enhanced Harris hawks optimization with multi-strategy for global optimization tasks. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2021.115499>
5. Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, *96*, 120–133. <https://doi.org/10.1016/j.knsys.2015.12.022>
6. Yang X. S. (2009). Firefly algorithms for multimodal optimization. In *Proceedings of stochastic algorithms: Foundations and applications, Berlin, Heidelberg* (pp. 169–178).
7. Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, *89*, 228–249. <https://doi.org/10.1016/j.knsys.2015.07.006>
8. Yang, X., Li, W., Su, L., Wang, Y., & Yang, A. (2020). An improved evolution fruit fly optimization algorithm and its application. *Neural Computing and Applications*, *32*, 9897–9914. <https://doi.org/10.1007/s00521-019-04512-2>
9. Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems-The International Journal of Esience*, *97*, 849–872. <https://doi.org/10.1016/j.future.2019.02.028>
10. Li, S. M., Chen, H. L., Wang, M. J., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems-the International Journal of Esience*, *111*, 300–323. <https://doi.org/10.1016/j.future.2020.03.055>
11. Kang, H., Bei, F., Shen, Y., Sun, X., & Chen, Q. (2021). A diversity model based on dimension entropy and its application to swarm intelligence algorithm. *Entropy*. <https://doi.org/10.3390/e23040397>
12. Hu, J., Wu, H., Zhong, B., & Xiao, R. (2020). Swarm intelligence-based optimisation algorithms: An overview and future research issues. *International Journal of Automation and Control*, *14*, 656–693. <https://doi.org/10.1504/IJAAC.2020.110077>
13. Anam S. & Fitriah Z. (2021). Early blight disease segmentation on tomato plant using K-means algorithm with swarm intelligence-based algorithm. *International Journal of Mathematics and Computer Science*, *16*, 1217–1228. <https://ijmcs.future-in-tech.net/16.4/R-Anam.pdf>
14. Al-Mousawi, A. J. (2021). Wireless communication networks and swarm intelligence. *Wireless Networks*, *27*, 1755–1782. <https://doi.org/10.1007/s11276-021-02545-x>
15. Kumar, G., Anwar, A., Dikshit, A., Poddar, A., Soni, U., & Song, W. K. (2022). Obstacle avoidance for a swarm of unmanned aerial vehicles operating on particle swarm optimization: A swarm intelligence approach for search and rescue missions. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, *44*, 56. <https://doi.org/10.1007/s40430-022-03362-9>
16. Xiang, T. (2020). Multi-scale feature fusion based on swarm intelligence collaborative learning for full-stage anti-interference object tracking. *Journal of Ambient Intelligence and Humanized Computing*. <https://doi.org/10.1007/s12652-019-01671-x>
17. Tunay, M., Pashaei, E., & Pashaei, E. (2022). Hybrid hypercube optimization search algorithm and multilayer perceptron neural network for medical data classification. *Computational Intelligence and Neuroscience*, *2022*, 16. <https://doi.org/10.1155/2022/1612468>
18. Bharanidharan, N., & Rajaguru, H. (2020). Performance enhancement of swarm intelligence techniques in dementia classification using dragonfly-based hybrid algorithms. *International Journal of Imaging Systems and Technology*, *30*, 57–74. <https://doi.org/10.1002/ima.22365>
19. Bach, H. N., Xue, B., & Zhang, M. (2020). A survey on swarm intelligence approaches to feature selection in data mining. *Swarm*

- and Evolutionary Computation. <https://doi.org/10.1016/j.swevo.2020.100663>
20. Nautiyal, B., Prakash, R., Vimal, V., Liang, G., & Chen, H. (2021). Improved Salp swarm algorithm with mutation schemes for solving global optimization and engineering problems. *Engineering with Computers*. <https://doi.org/10.1007/s00366-020-01252-z>
 21. Yıldız, B., Pholdee, N., Bureerat, S., Erdaş, M., Yildiz, A., & Sait, S. (2021). Comparison of the political optimization algorithm, the Archimedes optimization algorithm and the Levy flight algorithm for design optimization in industry. *Materials Testing*, 63, 356–359. <https://doi.org/10.1515/mt-2020-0053>
 22. Yildiz, B. S., Patel, V., Pholdee, N., Sait, S. M., Bureerat, S., & Yildiz, A. R. (2021). Conceptual comparison of the ecogeography-based algorithm, equilibrium algorithm, marine predators algorithm and slime mold algorithm for optimal product design. *Materials Testing*, 63, 336–340. <https://doi.org/10.1515/mt-2020-0049>
 23. Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, 95, 51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
 24. Wang, M., & Chen, H. (2020). Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis. *Applied Soft Computing*. <https://doi.org/10.1016/j.asoc.2019.105946>
 25. Chen, H. L., Xu, Y. T., Wang, M. J., & Zhao, X. H. (2019). A balanced whale optimization algorithm for constrained engineering design problems. *Applied Mathematical Modelling*, 71, 45–59. <https://doi.org/10.1016/j.apm.2019.02.004>
 26. Abdel-Basset, M., Mohamed, R., El-Fergany, A., Askar, S. S., & Abouhawwash, M. (2021). Efficient ranking-based whale optimizer for parameter extraction of three-diode photovoltaic model: Analysis and validations. *Energies*, 14, 1–21. <https://doi.org/10.3390/en14133729>
 27. Peng, H., Wen, W. S., Tseng, M. L., & Li, L. L. (2021). A cloud load forecasting model with nonlinear changes using whale optimization algorithm hybrid strategy. *Soft Computing*, 25, 10205–10220. <https://doi.org/10.1007/s00500-021-05961-5>
 28. Chakraborty, S., Kumar, S. A., Sharma, S., Mirjalili, S., & Chakraborty, R. (2021). A novel enhanced whale optimization algorithm for global optimization. *Computers and Industrial Engineering*. <https://doi.org/10.1016/j.cie.2020.107086>
 29. Ye, X. J., Liu, W., Li, H., Wang, M. J., Chi, C., Liang, G. X., Chen, H. L., & Huang, H. L. (2021). Modified whale optimization algorithm for solar cell and PV module parameter identification. *Complexity*, 2021, 1–23. <https://doi.org/10.1155/2021/8878686>
 30. Mostafa, A., Hassanien, A. E., Houseni, M., & Hefny, H. (2017). Liver segmentation in MRI images based on whale optimization algorithm. *Multimedia Tools and Applications*, 76, 24931–24954. <https://doi.org/10.1007/s11042-017-4638-5>
 31. Yang, C., & Wang, Y. Z. (2019). Inversion of the surface duct from radar sea clutter using the improved whale optimization algorithm. *Electromagnetics*, 39, 611–627. <https://doi.org/10.1080/02726343.2019.1675443>
 32. Hassib, E. M., El-Desouky, A. I., Labib, L. M., & El-kenawy, E. S. M. (2020). WOA plus BRNN: An imbalanced big data classification framework using Whale optimization and deep neural network. *Soft Computing*, 24, 5573–5592. <https://doi.org/10.1007/s00500-019-03901-y>
 33. Sayed, G. I., Darwish, A., & Hassanien, A. E. (2018). A new chaotic whale optimization algorithm for features selection. *Journal of Classification*, 35, 300–344. <https://doi.org/10.1007/s00357-018-9261-2>
 34. Tripathi, A. K., Mittal, H., Saxena, P., & Gupta, S. (2021). A new recommendation system using map-reduce-based tournament empowered Whale optimization algorithm. *Complex and Intelligent Systems*, 7, 297–309. <https://doi.org/10.1007/s40747-020-00200-0>
 35. Hussien, A. G., Hassanien, A. E., Houssein, E. H., Amin, M., & Azar, A. T. (2020). New binary whale optimization algorithm for discrete optimization problems. *Engineering Optimization*, 52, 945–959. <https://doi.org/10.1080/0305215x.2019.1624740>
 36. Chakraborty, S., Saha, A. K., Chakraborty, R., & Saha, M. (2021). An enhanced whale optimization algorithm for large scale optimization problems. *Knowledge-Based Systems*. <https://doi.org/10.1016/j.knsys.2021.107543>
 37. Luo, J., Chen, H. L., Heidari, A. A., Xu, Y. T., Zhang, Q., & Li, C. Y. (2019). Multi-strategy boosted mutative whale-inspired optimization approaches. *Applied Mathematical Modelling*, 73, 109–123. <https://doi.org/10.1016/j.apm.2019.03.046>
 38. Saha, N., & Panda, S. (2022). Cosine adapted modified whale optimization algorithm for control of switched reluctance motor. *Computational Intelligence*, 38, 978–1017. <https://doi.org/10.1111/coin.12310>
 39. Tu, J., Chen, H., Liu, J., Heidari, A. A., Zhang, X., Wang, M., Ruby, R., & Pham, Q.-V. (2021). Evolutionary biogeography-based whale optimization methods with communication structure: Towards measuring the balance. *Knowledge-Based Systems*. <https://doi.org/10.1016/j.knsys.2020.106642>
 40. Heidari, A. A., Aljarah, I., Faris, H., Chen, H., Luo, J., & Mirjalili, S. (2020). An enhanced associative learning-based exploratory whale optimizer for global optimization. *Neural Computing and Applications*, 32, 5185–5211. <https://doi.org/10.1007/s00521-019-04015-0>
 41. Chen, H. L., Yang, C. J., Heidari, A. A., & Zhao, X. H. (2020). An efficient double adaptive random spare reinforced whale optimization algorithm. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2019.113018>
 42. Chakraborty, S., Saha, A. K., Sharma, S., Chakraborty, R., & Debnath, S. (2021). A hybrid whale optimization algorithm for global optimization. *Journal of Ambient Intelligence and Humanized Computing*. <https://doi.org/10.1007/s12652-021-03304-8>
 43. Zhao, D., Liu, L., Yu, F. H., Heidari, A. A., Wang, M. J., Oliva, D., Muhammad, K., & Chen, H. L. (2021). Ant colony optimization with horizontal and vertical crossover search: Fundamental visions for multi-threshold image segmentation. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2020.114122>
 44. Elhosseini, M. A., Haikal, A. Y., Badawy, M., & Khashan, N. (2019). Biped robot stability based on an A-C parametric whale optimization algorithm. *Journal of Computational Science*, 31, 17–32. <https://doi.org/10.1016/j.jocs.2018.12.005>
 45. Chakraborty, S., Sharma, S., Saha, A. K., & Chakraborty, S. (2021). SHADE-WOA: A metaheuristic algorithm for global optimization. *Applied Soft Computing*. <https://doi.org/10.1016/j.asoc.2021.107866>
 46. Wang, W. L., Li, W. K., Wang, Z., & Li, L. (2019). Opposition-based multi-objective whale optimization algorithm with global grid ranking. *Neurocomputing*, 341, 41–59. <https://doi.org/10.1016/j.neucom.2019.02.054>
 47. Yang, L., & Chen, H. X. (2019). Fault diagnosis of gearbox based on RBF-PF and particle swarm optimization wavelet neural network. *Neural Computing and Applications*, 31, 4463–4478. <https://doi.org/10.1007/s00521-018-3525-y>
 48. Got, A., Moussaoui, A., & Zouache, D. (2021). Hybrid filter-wrapper feature selection using whale optimization algorithm: A multi-objective approach. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2021.115312>
 49. Abdel-Basset, M., Mohamed, R., AbdelAziz, N. M., & Abouhawwash, M. (2022). HWOA: A hybrid whale optimization algorithm with a novel local minima avoidance method for multi-level thresholding color image segmentation. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2021.116145>

50. Qi, A., Zhao, D., Yu, F., Heidari, A. A., Chen, H., & Xiao, L. (2022). Directional mutation and crossover for immature performance of whale algorithm with application to engineering optimization. *Journal of Computational Design and Engineering*, 9, 519–563. <https://doi.org/10.1093/jcde/qwac014>
51. Wang, G., Gui, W., Liang, G., Zhao, X., Wang, M., Mafarja, M., Turabieh, H., Xin, J., Chen, H., Ma, X., & Sun, Y. (2021). Spiral motion enhanced elite whale optimizer for global tasks. *Complexity*, 2021, 1–33. <https://doi.org/10.1155/2021/8130378>
52. Huynh-Thu, Q., & Ghanbari, M. (2008). Scope of validity of PSNR in image/video quality assessment. *Institution of Engineering and Technology*, 44(13), 800–801.
53. Zhou, W., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13, 600–612. <https://doi.org/10.1109/TIP.2003.819861>
54. Zhang, L., Zhang, L., Mou, X., & Zhang, D. (2011). FSIM: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 20, 2378–2386. <https://doi.org/10.1109/TIP.2011.2109730>
55. Rahnamayan S., Tizhoosh H. R. & Salama M. M. A. (2007). Quasi-oppositional differential evolution. In *Proceedings of 2007 IEEE Congress on Evolutionary Computation* (pp. 2229–2236).
56. Tizhoosh H. R. (2005). Opposition-based learning: A new scheme for machine intelligence. In *Proceedings of International Conference on Computational Intelligence for Modeling Control and Automation, Vienna, Austria* (pp. 695–701).
57. Wan, J., Chen, H., Li, T., Yuan, Z., Liu, J., & Huang, W. (2021). Interactive and complementary feature selection via fuzzy multi-granularity uncertainty measures. *IEEE Transactions on Cybernetics*. <https://doi.org/10.1109/TCYB.2021.3112203>
58. Agarwal, M., & Srivastava, G. M. S. (2021). Opposition-based learning inspired particle swarm optimization (OPSO) scheme for task scheduling problem in cloud computing. *Journal of Ambient Intelligence and Humanized Computing*, 12, 9855–9875. <https://doi.org/10.1007/s12652-020-02730-4>
59. Deng, W., Shang, S. F., Cai, X., Zhao, H. M., Song, Y. J., & Xu, J. J. (2021). An improved differential evolution algorithm and its application in optimization problem. *Soft Computing*, 25, 5277–5298. <https://doi.org/10.1007/s00500-020-05527-x>
60. Jiao, S., Chong, G., Huang, C., Hu, H., Wang, M., Heidari, A. A., Chen, H., & Zhao, X. (2020). Orthogonally adapted Harris hawks optimization for parameter estimation of photovoltaic models. *Energy*. <https://doi.org/10.1016/j.energy.2020.117804>
61. Wang, H., Rahnamayan, S., Sun, H., & Omran, M. G. H. (2013). Gaussian bare-bones differential evolution. *IEEE Transactions on Cybernetics*, 43, 634–647. <https://doi.org/10.1109/tsmcb.2012.2213808>
62. Kennedy J. (2003). Bare bones particle swarms. In *Proceedings of IEEE swarm intelligence symposium* (pp. 80–87).
63. Storn, R., & Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11, 341–359. <https://doi.org/10.1023/A:1008202821328>
64. Pan, W.-T. (2012). A new fruit fly optimization algorithm: Taking the financial distress model as an example. *Knowledge-Based Systems*, 26, 69–74. <https://doi.org/10.1016/j.knsys.2011.07.001>
65. Kennedy J. & Eberhart R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95—international conference on neural networks* (vol.1944, pp. 1942–1948).
66. Yang, Y. T., Chen, H. L., Heidari, A. A., & Gandomi, A. H. (2021). Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2021.114864>
67. Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7, 1–30.
68. Arora, S., & Anand, P. (2019). Binary butterfly optimization approaches for feature selection. *Expert Systems with Applications*, 116, 147–160. <https://doi.org/10.1016/j.eswa.2018.08.051>
69. Thaher, T., Chantar, H., Too, J., Mafarja, M., Turabieh, H., & Houssein, E. H. (2022). Boolean particle swarm optimization with various evolutionary population dynamics approaches for feature selection problems. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2022.116550>
70. Takieldean, A. E., El-kenawy, E.-S.M., Hadwan, M., & Zaki, R. M. (2022). Dipper throated optimization algorithm for unconstrained function and feature selection. *Cmc-Computers Materials and Continua*, 72, 1465–1481. <https://doi.org/10.32604/cmc.2022.026026>
71. Qaraad, M., Amjad, S., Hussein, N. K., & Elhosseini, M. A. (2022). Large scale salp-based grey wolf optimization for feature selection and global optimization. *Neural Computing and Applications*, 34, 8989–9018. <https://doi.org/10.1007/s00521-022-06921-2>
72. Kitonyi, P. M., & Segerer, D. R. (2021). Hybrid gradient descent grey wolf optimizer for optimal feature selection. *Biomed Research International*. <https://doi.org/10.1155/2021/2555622>
73. Dua, D. & Graff, C. (2019). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. <http://archive.ics.uci.edu/ml>
74. Nadimi-Shahraki, M. H., Banaie-Dezfouli, M., Zamani, H., Taghian, S., & Mirjalili, S. (2021). B-MFO: A binary moth-flame optimization for feature selection from medical datasets. *Computers*, 10, 136. <https://doi.org/10.3390/computers10110136>
75. Abdel-Basset, M., Mohamed, R., Sallam, K. M., Chakraborty, R. K., & Ryan, M. J. (2021). BSMA: A novel metaheuristic algorithm for multi-dimensional knapsack problems: Method and comprehensive analysis. *Computers and Industrial Engineering*. <https://doi.org/10.1016/j.cie.2021.107469>
76. Hashim, F. A., & Hussien, A. G. (2022). Snake optimizer: A novel meta-heuristic optimization algorithm. *Knowledge-Based Systems*, 242, 108320. <https://doi.org/10.1016/j.knsys.2022.108320>
77. Kumar, V., & Kumar, D. (2020). Binary whale optimization algorithm and its application to unit commitment problem. *Neural Computing and Applications*, 32, 2095–2123. <https://doi.org/10.1007/s00521-018-3796-3>
78. Chen, Y., Xie, W., & Zou, X. (2015). A binary differential evolution algorithm learning from explored solutions. *Neurocomputing*, 149, 1038–1047. <https://doi.org/10.1016/j.neucom.2014.07.030>
79. Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4, 40–79. <https://doi.org/10.1214/09-ss054>
80. Hu, J., Chen, H. L., Heidari, A. A., Wang, M. J., Zhang, X. Q., Chen, Y., & Pan, Z. F. (2021). Orthogonal learning covariance matrix for defects of grey wolf optimizer: Insights, balance, diversity, and feature selection. *Knowledge-Based Systems*. <https://doi.org/10.1016/j.knsys.2020.106684>
81. Chen, Y., Wang, M., Heidari, A. A., Shi, B., Hu, Z., Zhang, Q., Chen, H., Mafarja, M., & Turabieh, H. (2022). Multi-threshold image segmentation using a multi-strategy shuffled frog leaping algorithm. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2022.116511>
82. Abutaleb, A. S. (1989). Automatic thresholding of gray-level pictures using two-dimensional entropy. *Computer Vision, Graphics, and Image Processing*, 47, 22–32. [https://doi.org/10.1016/0734-189X\(89\)90051-0](https://doi.org/10.1016/0734-189X(89)90051-0)
83. Ji, W., & He, X. (2021). Kapur's entropy for multilevel thresholding image segmentation based on moth-flame optimization.

- Mathematical Biosciences and Engineering*, 18, 7110–7142. <https://doi.org/10.3934/mbe.2021353>
84. Wang, S., Sun, K., Zhang, W., & Jia, H. (2021). Multilevel thresholding using a modified ant lion optimizer with opposition-based learning for color image segmentation. *Mathematical Biosciences and Engineering*, 18, 3092–3143. <https://doi.org/10.3934/mbe.2021155>
 85. Zhao, D., Liu, L., Yu, F., Heidari, A. A., Wang, M., Liang, G., Muhammad, K., & Chen, H. (2021). Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D Kapur entropy. *Knowledge-Based Systems*. <https://doi.org/10.1016/j.knsys.2020.106510>
 86. Cai, J., Luo, T., Xu, G., & Tang, Y. (2022). A Novel biologically inspired approach for clustering and multi-level image thresholding: Modified Harris hawks optimizer. *Cognitive Computation*, 14, 955–969. <https://doi.org/10.1007/s12559-022-09998-y>
 87. Chakraborty, S., Saha, A. K., Nama, S., & Debnath, S. (2021). COVID-19 X-ray image segmentation by modified whale optimization algorithm with population reduction. *Computers in Biology and Medicine*, 139, 104984. <https://doi.org/10.1016/j.compbiomed.2021.104984>
 88. Wang, X., Li, Z., Kang, H., Huang, Y., & Gai, D. (2021). Medical image segmentation using PCNN based on multi-feature grey wolf optimizer bionic algorithm. *Journal of Bionic Engineering*, 18, 711–720. <https://doi.org/10.1007/s42235-021-0049-4>
 89. Sharma, S., Saha, A. K., Majumder, A., & Nama, S. (2021). MPBOA—a novel hybrid butterfly optimization algorithm with symbiosis organisms search for global optimization and image segmentation. *Multimedia Tools and Applications*, 80, 12035–12076. <https://doi.org/10.1007/s11042-020-10053-x>
 90. Chakraborty, S., Sharma, S., Saha, A. K., & Saha, A. (2022). A novel improved whale optimization algorithm to solve numerical optimization and real-world applications. *Artificial Intelligence Review*, 55, 4605–4716. <https://doi.org/10.1007/s10462-021-10114-z>
 91. Zhao, S., Wang, P., Heidari, A. A., Chen, H., He, W., & Xu, S. (2021). Performance optimization of salp swarm algorithm for multi-threshold image segmentation: Comprehensive study of breast cancer microscopy. *Computers in Biology and Medicine*, 139, 105015. <https://doi.org/10.1016/j.compbiomed.2021.105015>
 92. Cohen J. P., Morrison P., Dao L., Roth K., Duong T. Q. & Ghassemi M. 2020. COVID-19 image data collection: Prospective predictions are the future. <https://doi.org/10.48550/arXiv.2006.11988>
 93. Ewees, A. A., Al-qaness, M. A. A., & Abd, E. M. (2021). Enhanced salp swarm algorithm based on firefly algorithm for unrelated parallel machine scheduling with setup times. *Applied Mathematical Modelling*, 94, 285–305. <https://doi.org/10.1016/j.apm.2021.01.017>
 94. Wang, B. C., Li, H. X., Zhang, Q. F., & Wang, Y. (2021). Decomposition-based multiobjective optimization for constrained evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51, 574–587. <https://doi.org/10.1109/tsmc.2018.2876335>
 95. Naderipour, A., Abdullah, A., Marzbali, M. H., & Arabi, N. S. (2022). An improved corona-virus herd immunity optimizer algorithm for network reconfiguration based on fuzzy multi-criteria approach. *Expert Systems with Applications*, 187, 115914. <https://doi.org/10.1016/j.eswa.2021.115914>
 96. Li, F., Su, Z., & Wang, G. (2022). An effective dynamic immune optimization control for the wastewater treatment process. *Environmental Science and Pollution Research*. <https://doi.org/10.1007/s11356-021-17505-3>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.