Hindawi
Computational Intelligence and Neuroscience
Volume 2020, Article ID 6858541, 24 pages
https://doi.org/10.1155/2020/6858541

Research Article

Dynamic Multi-Swarm Differential Learning Quantum Bird Swarm Algorithm and Its Application in Random Forest Classification Model

Jiangnan Zhang, Kewen Xia, Ziping He, and Shurui Fan

School of Electronic and Information Engineering, Hebei University of Technology, Tianjin 300401, China

Correspondence should be addressed to Kewen Xia; kwxia@hebut.edu.cn

Received 14 August 2019; Revised 21 June 2020; Accepted 24 June 2020; Published 7 August 2020

Academic Editor: Raşit Köker

Bird swarm algorithm is one of the swarm intelligence algorithms proposed recently. However, the original bird swarm algorithm has some drawbacks, such as easy to fall into local optimum and slow convergence speed. To overcome these short-comings, a dynamic multi-swarm differential learning quantum bird swarm algorithm which combines three hybrid strategies was established. First, establishing a dynamic multi-swarm bird swarm algorithm and the differential evolution strategy was adopted to enhance the randomness of the foraging behavior's movement, which can make the bird swarm algorithm have a stronger global exploration capability. Next, quantum behavior was introduced into the bird swarm algorithm for more efficient search solution space. Then, the improved bird swarm algorithm is used to optimize the number of decision trees and the number of predictor variables on the random forest classification model. In the experiment, the 18 benchmark functions, 30 CEC2014 functions, and the 8 UCI datasets are tested to show that the improved algorithm and model are very competitive and outperform the other algorithms and models. Finally, the effective random forest classification model was applied to actual oil logging prediction. As the experimental results show, the three strategies can significantly boost the performance of the bird swarm algorithm and the proposed learning scheme can guarantee a more stable random forest classification model with higher accuracy and efficiency compared to others.

1. Introduction

The concept of swarm intelligence was first proposed by Hackwood and Beni in 1992 [1]. Swarm intelligence algorithms have been proved that it can solve nondifferentiable problems, NP-hard problems, and difficult nonlinear problems which the traditional techniques cannot solve. For this reason, swarm intelligence algorithms are hotly researched in computer science and have been updated from generation to generation. For classic swarm intelligence algorithms, particle swarm optimization (PSO) [2] is used to define the basic principle and equations of the swarm intelligence algorithms. In recent years, many new swarm intelligence algorithms have been proposed, such as artificial bee colony (ABC) algorithm [3] which is inspired by the stock of food location behavior of bees. Artificial fish school algorithm (AFSA) [4] and firefly algorithm (FA) [5] are inspired by the foraging process of fish and firefly, and cat swarm optimization (CSO) [6] is developed based on vigilance and foraging behavior of cats in nature. According to the foraging behavior, vigilance behavior, and flight behavior of the bird swarms in nature, Meng et al. proposed a novel swarm intelligence algorithm called bird swarm algorithm (BSA) [7]. Meanwhile, due to these advantages above, swarm intelligence algorithms have been applied to optimize various fields, such as PSO for mutation testing problems [8], genetic algorithm (GA) for convolutional neural networks parameters [9], FA for convolutional neural network problems [10], and whale optimization algorithm (WOA) for cloud computing environments [11]. So, BSA which will be used in this paper has been widely applied to engineering optimization problems.

However, the original swarm intelligence algorithms have limitations in solving some practical problems. Hybrid strategy which is one of the main research directions to improve the performance of swarm intelligence algorithms has become a research hotspot in machine learning. Tuba and Bacanin [12] modified the exploitation process of the original seeker optimization algorithm (SOA) approach by hybridizing it with FA which overcame shortcomings and outperformed other algorithms. Strumberger et al. [13] also has proposed dynamic search tree growth algorithm (TGA) and hybridized elephant herding optimization (EHO) with ABC, and the simulation results have shown that the proposed approach was viable and effective. Yang [14] analyzed swarm intelligence algorithms by using differential evolution, dynamic systems, self-organization, and Markov chain framework. The discussions demonstrate that the hybrid algorithms have some advantages over traditional algorithms. Bacanin and Tuba [15] proposed a modified ABC based on GA, and the obtained results show that the hybrid ABC is able to provide competitive results and outperform other counterparts. Liu et al. [16] presented a multistrategy brain storm optimization (BSO) with dynamic parameter adjustment which is more competitive than other related algorithms. Peng et al. [17] has proposed FA with luciferase inhibition mechanism to improve the effectiveness of selection. The simulation results have shown that the proposed approach has the best performance in some complex functions. Peng et al. [18] also developed a hybrid approach, which is using the best neighbor-guided solution search strategy to search ABC algorithm. The experimental results indicate that the proposed ABC is very competitive and outperforms the other algorithms. It can be seen that the hybrid strategy is a strategy to successfully improve the swarm intelligence algorithm, so the BSA algorithm will be improved by hybrid strategy in this paper.

Similarly, BSA can also be applied to multiple fields, especially in the field of parameter estimation, and hybrid strategy is also the improvement method for BSA. In 2017, Xu et al. [19] proposed improved boundary BSA (IBBSA) for chaotic system optimization of the Lorenz system and the coupling motor system. However, the improved boundary learning strategy has randomness, which makes IBBSA generalization performance not high. Yang and Liu [20] introduced the dynamic weight into the foraging formula of BSA (IBSA) which provides a solution for problem that anti-same-frequency interference of shipborne radar. The results have shown that the dynamic weight is just introduced into the foraging formula of BSA, but IBSA ignored the impact of population initialization. Wang et al. [21] designed a strategy named "disturbing the local optimum" for helping the original BSA converge to the global optimal solution faster and more stably. However, "disturbing the local optimum" also has randomness, which makes the generalization performance of improved BSA not very well.

Like many swarm intelligence algorithms, BSA is also faced with the problem of being trapped in local optima and slow convergence. These disadvantages limit the wider application of BSA. In this paper, a dynamic multi-swarm differential learning quantum BSA called DMSDL-QBSA is proposed, which introduced three hybrid strategies into the original BSA to improve its effectiveness. Motivated by the defect of insufficient generalization ability in the literature [19, 21], we will first establish a dynamic multi-swarm bird swarm algorithm (DMS-BSA) and merge the differential evolution operator into each sub-swarm of the DMS-BSA, and it improves the local search capability and global search capability of foraging behavior. Second, according to the contempt for the impact of population initialization in the literature [20], they used quantum behavior to optimize the particle swarm optimization in order to obtain a good ability to jump out of global optimum; we will use the quantum system to initialize the search space of the bird. Consequently, it improves the convergence rate of the whole population and avoids BSA into a local optimum. In order to validate effectiveness of the proposed method, we have evaluated the performance of DMSDL-QBSA on classical benchmark functions and CEC2014 functions including unimodal and multimodal functions in comparison with the state-of-the-art methods and new popular algorithms. The experimental results have shown that the three improvement strategies are able to significantly boost the performance of BSA.

Based on the DMSDL-QBSA, an effective hybrid random forest (RF) model for actual oil logging prediction is established, called DMSDL-QBSA-RF approach. RF has the characteristics of being nonlinear and anti-interference [22]. In addition, it can decrease the possibility of overfitting which often occurs in actual logging. RF has been widely used in various classification problems, but it has not yet been applied to the field of actual logging. Parameter estimation is a prerequisite to accomplish the RF classification model. The two key parameters of RF are the number of decision trees and the number of predictor variables; the former is called $[-100, 100]^D$, and the latter is called m_{try}. Meanwhile, parameter estimation of the model is a complex optimization problem that traditional methods might fail to solve. Many works have proposed to use swarm intelligence algorithms to find the best parameters of the RF model. Ma and Fan [23] adopted AFSA and PSO to optimize the parameters of the RF. Hou et al. [24] used the DE to obtain an optimal set of initial parameters for RF. Liu et al. [25] compared genetic algorithms, simulated annealing, and hill climbing algorithms to optimize the parameters of the RF. From these papers, we can see that metaheuristic algorithm must be suitable for this problem. In this study, the DMSDL-QBSA was used to optimize the two key parameters that can improve the accuracy without overfitting for RF. When investigating the performance of the DMSDL-QBSA-RF classification model compared with 3 swarm intelligence algorithm-based RF methods, 8 two-dimensional UCI datasets are applied. As the experimental results show, the proposed learning scheme can guarantee a more stable RF classification model with higher predictive accuracy compared to other counterparts. The rest of the paper is organized as follows:

(i) In order to achieve a better balance between efficiency and velocity for BSA, we have studied the effects of four different hybrid strategies of the dynamic multi-swarm method, differential evolution, and quantum behavior on the performance of BSA.

(ii) The proposed DMSDL-QBSA has successfully optimized n_{tree} and m_{try} setting problem of RF. The resulting hybrid classification model has been rigorously evaluated on oil logging prediction.

(iii) The proposed hybrid classification model delivers better classification performance and offers more accurate and faster results when compared to other swarm intelligence algorithm-based RF models.

2. Bird Swarm Algorithm and Its Improvement

2.1. Bird Swarm Algorithm Principle. BSA, as proposed by Meng et al. in 2015, is a new intelligent bionic algorithm based on multigroup and multisearch methods; it mimics the birds' foraging behavior, vigilance behavior, and flight behavior, and employs this swarm intelligence to solve the optimization problem. The bird swarm algorithm can be simplified by the five rules:

Rule 1: each bird can switch between vigilant behavior and foraging behavior, and both bird forages and keeps vigilance is mimicked as random decisions.

Rule 2: when foraging, each bird records and updates its previous best experience and the swarms' previous best experience with food patches. The experience can also be used to search for food. Instant sharing of social information is across the group.

Rule 3: when keeping vigilance, each bird tries to move towards the center of the swarm. This behavior may be influenced by disturbances caused by swarm competition. Birds with more stocks are more likely to be near swarm's centers than birds with lease stocks.

Rule 4: birds fly to another place regularly. When flying to another location, birds often switch between production and shrubs. The bird with the most stocks is the producer, and the bird with the least is a scrounger. Other birds with the highest and lowest reserves are randomly selected for producers and scroungers.

Rule 5: producers actively seek food. Scroungers randomly follow producers looking for food.

According to Rule 1, we define that the time interval of each bird flight behavior FQ, the probability of foraging behavior $P (P \in (0,1))$, and a uniform random number $\delta \in (0,1)$.

(1) Foraging behavior

If the number of iteration is less than FQ and $\delta \leq P$, the bird will be the foraging behavior. Rule 2 can be written mathematically as follows:

$$\begin{aligned} x_{i,j}^{t+1} &= x_{i,j}^{t} + \left(p_{i,j}^{t} - x_{i,j}^{t}\right) \times C \times \text{rand}(0,1) \\ &\quad + \left(g_{j}^{t} - x_{i,j}^{t}\right) \times S \times \text{rand}(0,1), \end{aligned} \tag{1}$$

where C and S are two positive numbers; the former is called cognitive accelerated coefficients, and the latter is called social accelerated coefficients. Here, $p_{i,j}$ is the i-th bird's best previous position and g_j is the best previous swarm's position.

(2) Vigilance behavior

If the number of iteration is less than FQ and $\delta > P$, the bird will be the vigilance behavior. Rule 3 can be written mathematically as follows:

$$\begin{aligned} x_{i,j}^{t+1} &= x_{i,j}^{t} + A_1\left(\text{mean}_j^{t} - x_{i,j}^{t}\right) \times \text{rand}(0,1) \\ &\quad + A_2\left(p_{k,j}^{t} - x_{i,j}^{t}\right) \times \text{rand}(-1,1), \end{aligned} \tag{2}$$

$$A_1 = a_1 \times \exp\left(-\frac{p\text{Fit}_i}{\text{sumFit} + \varepsilon} \times N\right), \tag{3}$$

$$A_2 = a_2 \times \exp\left(\left(\frac{p\text{Fit}_i - p\text{Fit}_k}{\left|p\text{Fit}_k - p\text{Fit}_i\right| + \varepsilon}\right) \times \frac{N \times p\text{Fit}_k}{\text{sumFit} + \varepsilon}\right), \tag{4}$$

where a_1 and a_2 are two positive constants in $[0,2]$, $p\text{Fit}_i$ is the best fitness value of i-th bird, and sumFit is the sum of the swarms' best fitness value. Here, ε, which is used to avoid zero-division error, is the smallest constant in the computer. mean_j denotes the j-th element of the whole swarm's average position.

(3) Flight behavior

If the number of iteration equals FQ, the bird will be the flight behavior which can be divided into the behaviors of the producers and scroungers by fitness. Rule 3 and Rule 4 can be written mathematically as follows:

$$x_{i,j}^{t+1} = x_{i,j}^{t} + \text{randn}(0,1) \times x_{i,j}^{t}, \tag{5}$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + \left(x_{k,j}^{t} - x_{i,j}^{t}\right) \times \text{FL} \times \text{rand}(0,1), \tag{6}$$

where FL $(\text{FL} \in [0,2])$ means that the scrounger would follow the producer to search for food.

2.2. The Bird Swarm Algorithm Based on Dynamic Multi-Swarm Method, Differential Evolution, and Quantum Behavior

2.2.1. The Foraging Behavior Based on Dynamic Multi-Swarm Method. Dynamic multi-swarm method has been widely used in real-world applications, because it is efficient and easy to implement. In addition, it is very common in the improvement of swarm intelligent optimization, such as coevolutionary algorithm [26], the framework of evolutionary algorithms [27], multiobjective particle swarm optimization [28], hybrid dynamic robust [29], and PSO algorithm [30, 31]. However, the PSO algorithm is easy to fall into the local optimum and its generalization performance is not high. Consequently, motivated by these literature studies, we will establish a dynamic multi-swarm bird swarm algorithm (DMS-BSA), and it improves the local search capability of foraging behavior.

In DMS-PSO, the whole population is divided into many small swarms, which are often regrouped by using various reorganization plans to exchange information. The velocity update strategy is

$$v_{i,j}^{t+1} = \omega v_{i,j}^{t} + c_2 r_2\left(l_{i,j}^{t} - x_{i,j}^{t}\right), \tag{7}$$

where $l_{i,j}$ is the best historical position achieved within the local community of the i-th particle.

According to the characteristic of equation (1), we can see that the foraging behavior formula of BSA is similar to the particle velocity update formula of PSO. So, according to the characteristic of equation (7), we can get the improved foraging behavior formula as follows:

$$x_{i,j}^{t+1} = x_{i,j}^{t} + \left(GV_{i,j}^{t} - x_{i,j}^{t}\right) \times C \times \text{rand}(0,1), \tag{8}$$

where GV is called the guiding vector.

The dynamic multi-swarm method is used to improve the local search capability, while the guiding vector GV can enhance the global search capability of foraging behavior. Obviously, we need to build a good guiding vector.

2.2.2. The Guiding Vector Based on Differential Evolution. Differential evolution (DE) is a powerful evolutionary algorithm with three differential evolution operators for solving the tough global optimization problems [32]. Besides, DE has got more and more attention of scholars to evolve and improve in evolutionary computation, such as hybrid multiple crossover operations [33] and proposed DE/neighbor/1 [34], due to its excellent global search capability. From these literature studies, we can see that DE has a good global search capability, so we will establish the guiding vector GV based on differential evolution operator to improve the global search capability of foraging behavior. The detailed implementation of GV is presented as follows:

(1) Differential mutation

According to the characteristic of equation (8), the "DE/best/1", "DE/best/2," and "DE/current-to-best/1" mutation strategies are suitable. In the experiments of the literature [31], they showed that the "DE/best/1" mutation strategy is the most suitable in DMS-PSO, so we choose this mutation strategy in BSA. And the "DE/lbest/1" mutation strategy can be written as follows:

DE/lbest/1 :

$$v_{i,j}^{t} = l_{i,j}^{t} + F \times \left(p_{r1,j}^{t} - p_{r2,j}^{t}\right). \tag{9}$$

Note that some components of the mutant vector $v_{i,j}$ may violate predefined boundary constraints. In this case, the boundary processing is used. It can be expressed as follows:

$$v_{i,j}^{t} = \begin{cases} x_{\text{lb}}, & v_{i,j}^{t} < x_{\text{lb}}, \\ x_{\text{ub}}, & v_{i,j}^{t} \geq x_{\text{ub}}. \end{cases} \tag{10}$$

(2) Crossover

After differential mutation, a binomial crossover operation exchanges some components of the mutant vector $v_{i,j}$ with the best previous position $p_{i,j}$ to generate the target vector $u_{i,j}$. The process can be expressed as

$$u_{i,j}^{t} = \begin{cases} v_{i,j}^{t}, & \text{rand} \leq \text{CR or } j = \text{rand}(i), \\ p_{i,j}^{t}, & \text{otherwise}. \end{cases} \tag{11}$$

(3) Selection

Because the purpose of BSA is to find the best fitness, a selection operation chooses a vector accompanied a better fitness to enter the next generation to generate the selection operator, namely, guiding vector GV. The process can be expressed as follows:

Choose a vector with better fitness to enter the next generation

$$\text{GV}_{i,j}^{t+1} = \begin{cases} u_{i,j}^{t}, & \text{fitness}\left(u_{i,j}^{t}\right) \leq \text{fitness}\left(x_{i,j}^{t}\right), \\ p_{i,j}^{t}, & \text{otherwise}. \end{cases} \tag{12}$$

2.2.3. The Initialization of Search Space Based on Quantum Behavior. Quantum behavior is a nonlinear and excellent superposition system. With its simple and effective characteristics and good performance in global optimization, it has been applied to optimize many algorithms, such as particle swarm optimization [35] and pigeon-inspired optimization algorithm [36]. Consequently, according to the literature studies and its excellent global optimization performance, we use the quantum system to initialize the search space of the bird.

Quantum-behaved particle position can be written mathematically as follows:

$$x^{t+1} = p^{t} \pm \frac{L^{t}}{2} \ln \frac{1}{u}, \tag{13}$$

$$p^{t} = c_1 \times p\text{best}^{t} + \left(1 - c_1\right) \times g\text{best}^{t}, \tag{14}$$

$$L^{t} = 2\beta\left|P^{t} - x^{t}\right|. \tag{15}$$

According to the characteristics of equations (13)–(15), we can get the improved search space initialization formula as follows:

$$x_i^t = \text{lb} + (\text{ub} - \text{lb}) \times \text{rand}, \tag{16}$$

$$x_i^{t+1} = x_i^t \pm \left| m\text{best} - x_i^t \right| \times \ln \frac{1}{u}, \tag{17}$$

where β is a positive number, which can be, respectively, called as a contraction expansion factor. Here, x_i^t is the position of the particle at the previous moment and mbest is the average value of the best previous positions of all the birds (Algorithm 1).

2.2.4. Procedures of the DMSDL-QBSA. In Sections 2.2.1–2.2.3, in order to improve the local search capability and the global search capability on BSA, this paper has improved the BSA in three parts:

(1) In order to improve the local search capability of foraging behavior on BSA, we put forward equation (8) based on the dynamic multi-swarm method.

(2) In order to get the guiding vector to improve the global search capability of foraging behavior on BSA, we put forward equations (9), (11), and (12) based on differential evolution.

(3) In order to expand the initialization search space of the bird to improve the global search capability on BSA, we put forward equations (16) and (17) based on quantum behavior.

Finally, the steps of DMSDL-QBSA can be shown in Algorithm 1.

2.3. Simulation Experiment and Analysis. This section presents the evaluation of DMSDL-QBSA using a series of experiments on benchmark functions and CEC2014 test functions. All experiments in this paper are implemented using the following: MATLAB R2014b; Win 7 (64-bit); Inter (R) Core (TM) i5-2450M; CPU @2.50 GHz; 4.00 GB RAM. To obtain fair results, all the experiments were conducted under the same conditions. The number of the population size is set as 30 in these algorithms. And each algorithm runs 30 times independently for each function.

2.3.1. Benchmark Functions and CEC 2014 Test Functions. When investigating the effective and universal performance of DMSDL-QBSA compared with several hybrid algorithms and popular algorithms, 18 benchmark functions and CEC2014 test functions are applied. In order to test the effectiveness of the proposed DMSDL-QBSA, 18 benchmark functions [37] are adopted, and all of which have an optimal value of 0. The benchmark functions and their searching ranges are shown in Table 1. In this test suite, $f_1 - f_9$ are unimodal functions. These unimodal functions are usually used to test and investigate whether the proposed algorithm has a good convergence performance. Then, $f_{10} - f_{18}$ are multimodal functions. These multimodal functions are used to test the global search capability of the proposed algorithm. The smaller the fitness value of functions, the better the algorithm performs. Furthermore, in order to better verify the comprehensive performance of DMSDL-QBSA in a more comprehensively manner, another 30 complex CEC2014 benchmarks are used. The CEC2014 benchmark functions are simply described in Table 2.

2.3.2. Parameter Settings. In order to verify the effectiveness and generalization of the proposed DMSDL-QBSA, the improved DMSDL-QBSA is compared with several hybrid algorithms. These algorithms are BSA [7], DE [32], DMSDL-PSO [31], and DMSDL-BSA. Another 5 popular intelligence algorithms, such as grey wolf optimizer (GWO) [38], whale optimization algorithm (WOA) [39], sine cosine algorithm (SCA) [40], grasshopper optimization algorithm (GOA) [41], and sparrow search algorithm (SSA) [42], are used to compare with DMSDL-QBSA. These algorithms represented state-of-the-art can be used to better verify the performance of DMSDL-QBSA in a more comprehensively manner. For fair comparison, the number of populations of all algorithms is set to 30, respectively, and other parameters of all algorithms are set according to their original papers. The parameter settings of these involved algorithms are shown in Table 3 in detail.

2.3.3. Comparison on Benchmark Functions with Hybrid Algorithms. According to Section 2.2, three hybrid strategies (dynamic multi-swarm method, DE, and quantum behavior) have been combined with the basic BSA method. When investigating the effectiveness of DMSDL-QBSA compared with several hybrid algorithms, such as BSA, DE, DMSDL-PSO, and DMSDL-BSA, 18 benchmark functions are applied. Compared with DMSDL-QBSA, quantum behavior dynamic is not used in the dynamic multi-swarm differential learning bird swarm algorithm (DMSDL-BSA). The number of function evaluations (FEs) is 10000. We selected two different dimension's sizes (Dim). Dim = 10 is the typical dimensions for the benchmark functions. And Dim = 2 is for RF has two parameters that need to be optimized, which means that the optimization function is 2-dimensional.

The fitness value curves of a run of several algorithms on about eight different functions are shown in Figures 1 and 2, where the horizontal axis represents the number of iterations and the vertical axis represents the fitness value. We can obviously see the convergence speeds of several different algorithms. The maximum value (Max), the minimum value (Min), the mean value (Mean), and the variance (Var) obtained by several benchmark algorithms are shown in Tables 4–7, where the best results are marked in bold. Table 4 and 5 show the performance of the several algorithms on unimodal functions when Dim = 10 and 2, and Table 6 and 7 show the performance of the several algorithms on multimodal functions when Dim = 10 and 2.

(1) Unimodal functions

From the numerical testing results on 8 unimodal functions in Table 4, we can see that DMSDL-QBSA can find the optimal solution for all unimodal functions and get the minimum value of 0 on f_1, f_2, f_3, f_7, and f_8. Both DMSDL-QBSA and DMSDL-

```
Variable setting: number of iterations:iter, bird positions: $X_i$, local optimum: $P_i$, global optimal position:gbest, global optimum:
    $P_{\text{gbest}}$,and fitness of sub-swarm optimal position: $P_{\text{lbest}}$;
Input: population size: $N$, dimension's size: $D$, number of function evaluations:iter_ max, the time interval of each bird flight behavior:
    FQ, the probability of foraging behavior: $P$, constant parameter: $c_1$, $c_2$, $a_1$, $a_2$, FL, iter = 0, and contraction expansion factor: $\beta$;
Output: global optimal position: gbest, fitness of global optimal position: $P_{\text{gbest}}$;
Begin
Initialize the positions of N birds using equations (16)-(17): $X_i\,(i = 1, 2, \ldots, N)$;
Calculated fitness: $F_i\,(i = 1, 2, \ldots, N)$; set $X_i$ to be $P_i$ and find $P_{\text{gbest}}$;
While $\text{iter} < \text{iter}_{\max} + 1$ do
/* Dynamic sub-swarm*/
Regroup $P_i$ and $X_i$ of the sub-swarms randomly;
Sort the $P_{\text{lbest}}$ and refine the first $[0.25 * N]$ best $P_{\text{lbest}}$;
update the corresponding $P_i$;
    /*Differential learning scheme*/
    For $i = 1$: $N$
        Construct lbest using DMS-BSA
        Differential evolution: construct $V$ using equations (9)-(10);
        Crossover: construct $U$ using equation (11);
        Selection: construct GV using equation (12);
    End For
    /*Birds position adjusting*/
    If $(t\%\text{FQ} \neq 0)$
        For $i = 1$: $N$
            If rand $< p$
                Foraging behavior: update the position $X_i$ of birds using equation (8);
            Else
                Vigilance behavior: update the position $X_i$ of birds using equation (2);
            End If
        End For
    Else
        Flight behavior is divided into producers and scroungers;
        For $i = 1$: $N$
            If $X_i$ is a producer
            Producers: update the position $X_i$ of birds using equation (5);
        Else
            Scroungers: update the position $X_i$ of birds using equation (6);
        End If
    End For
    End If
    Evaluate $f\,(X_i)$;
    Update gbest and $P_{\text{gbest}}$;
        iter = iter + 1;
    End While
End
```

Algorithm 1: DMSDL-QBSA.

BSA can find the minimum value. However, DMSDL-QBSA has the best mean value and variance on each function. The main reason is that DMSDL-QBSA has better population diversity during the initialization period. In summary, the DMSDL-QBSA has best performance on unimodal functions compared to the other algorithms when Dim = 10. Obviously, the DMSDL-QBSA has a relatively well convergence speed.

The evolution curves of these algorithms on four unimodal functions f_1, f_5, f_6, and f_9 are drawn in Figure 1. It can be detected from the figure that the curve of DMSDL-QBSA descends fastest in the number of iterations that are far less than 10000 times. For f_1, f_5, and f_9 case, DMSDL-QBSA has the fastest convergence speed compared with other algorithms. However, the original BSA got the worst solution because it is trapped in the local optimum prematurely. For function f_6, these algorithms did not find the value 0. However, the convergence speed of the DMSDL-QBSA is significantly faster than other algorithms in the early stage and the solution eventually found is the best. Overall, owing to enhance the diversity of population, DMSDL-QBSA has a relatively excellent convergence speed when Dim = 2.

Table 1: 18 benchmark functions.

Name	Test function	Range
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$
Schwefel P2.22	$f_2 = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$	$[-10, 10]^D$
Schwefel P1.2	$f_3(x) = \sum_{i=1}^{D} (\sum_{j}^{i} x_j)^2$	$[-100, 100]^D$
Generalized Rosenbrock	$f_4(x) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$[-100, 100]^D$
Step	$f_5(x) = \sum_{i=1}^{D} (\lvert x_i + 0.5\rvert)^2$	$[-100, 100]^D$
Noise	$f_6(x) = \sum_{i=1}^{D} i x_i^4 + \text{rand}[0, 1)$	$[-1.28, 1.28]^D$
SumSquares	$f_7(x) = \sum_{i=1}^{D} i x_i^2$	$[-10, 10]^D$
Zakharov	$f_8(x) = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5 i x_i)^2 + (\sum_{i=1}^{D} 0.5 i x_i)^4$	$[-10, 10]^D$
Schaffer	$f_9(x, y) = 0.5 + (\sin^2(x^2 - y^2) - 0.5/[1 + 0.001(x^2 - y^2)]^2)$	$[-10, 10]^D$
Generalized Schwefel 2.26	$f_{10}(x) = 418.9829\,D - \sum_{i=1}^{D} (x \sin \sqrt{\lvert x_i\rvert})$	$[-500, 500]^D$
Generalized Rastrigin	$f_{11}(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$
Ackley	$f_{12} = 20 + e - 20\exp(-0.2\sqrt{(1/D)\sum_{i=1}^{D} x_i^2}) - \exp((1/D)\sum_{i=1}^{D} \cos 2\pi x_i)$	$[-32, 32]^D$
Generalized Griewank	$f_{13}(x) = \sum_{i=1}^{D} (x_i^2/4000) - \prod_{i=1}^{D} \cos(x_i/\sqrt{i}) + 1$	$[-600, 600]^D$
Generalized Penalized 1	$f_{14}(x) = (\pi/D)\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2$ $[1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4).$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \le x_i \le a \\ k(-x_i - a)^m, & x_i < -a \end{cases}, \quad y_i = 1 + (1/4)x_i$	$[-50, 50]^D$
Generalized Penalized 2	$f_{15}(x) = (1/10)\{10\sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})]$ $+ (x_D - 1)^2 [1 + \sin^2(2\pi x_D)]\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$	$[-50, 50]^D$
Alpine	$f_{16}(x) = \sum_{i=1}^{D} \lvert x_i \sin x_i + 0.1 x_i\rvert$	$[-10, 10]^D$
Booth	$f_{17}(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	$[-10, 10]^D$
Levy	$f_{18}(x) = \sin^2(\pi x_1) + \sum_{i=1}^{D} \{(y_i - 1)^2 [1 + 10\sin^2(\pi y_i + 1)]\} + (y_i - 1)^2 [1 + 10\sin^2(2\pi y_D)]$	$[-10, 10]^D$

According to the results of Table 5, DMSDL-QBSA gets the best performance on these 8 unimodal functions when Dim = 2. DMSDL-QBSA finds the minimum value of 0 on f_1, f_2, f_3, f_4, f_5, f_7, f_8, and f_9. DMSDL-QBSA has better performance on f_2, f_4, and f_9 compared with DMSDL-BSA and DMSDL-PSO. DE does not perform well on these functions, but BSA performs relatively well on f_1, f_3, f_6, f_7, f_8, and f_9. The main reason is that BSA has a better convergence performance in the early search. Obviously, DMSDL-QBSA can find the best two parameters for RF that need to be optimized.

(2) Multimodal functions

From the numerical testing results on 8 multimodal functions in Table 6, we can see that DMSDL-QBSA can find the optimal solution for all multimodal functions and get the minimum value of 0 on f_{11}, f_{13}, and f_{16}. DMSDL-QBSA has the best performance on f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, and f_{16}. BSA works the best on f_{10}. DMSDL-PSO performs not very well. And DMSDL-QBSA has the best mean value and variance on most functions. The main reason is that DMSDL-QBSA has a stronger global exploration capability based on the dynamic multi-swarm method and differential evolution. In summary, the DMSDL-QBSA has relatively well performance on unimodal functions compared to the other algorithms when typical Dim = 10. Obviously, the DMSDL-QBSA has relatively well global search capability.

The evolution curves of these algorithms on four multimodal functions f_{12}, f_{13}, f_{17}, and f_{18} when Dim = 2 are depicted in Figure 2. We can see that DMSDL-QBSA can find the optimal solution in the same iteration. For f_{13} and f_{17} case, DMSDL-QBSA continues to decline. However, the original BSA and DE get parallel straight lines because of their poor global convergence ability. For functions f_{12} and f_{18}, although DMSDL-QBSA also trapped the local optimum, it find the minimum value compared to other algorithms. Obviously, the convergence speed of the DMSDL-QBSA is significantly faster than other algorithms in the early stage, and the solution eventually found is the best. In general, owing to enhance the diversity of population, DMSDL-QBSA has a relatively balanced global search capability when Dim = 2.

Furthermore, from the numerical testing results on nine multimodal functions in Table 7, we can see that DMSDL-QBSA has the best performance on f_{11}, f_{12}, f_{13}, f_{14}, f_{16}, f_{17}, and f_{18}. DMSDL-QBSA gets the minimum value of 0 on f_{11}, f_{13}, f_{16}, and f_{17}. BSA has got the minimum value of 0 on f_{11}, f_{13}, and f_{17}. DE also has not got the minimum value of 0

Table 2: Summary of the CEC′14 test functions.

Type	No.	Functions	$Fi^* = Fi\ (x^*)$
Unimodal functions	F_1	Rotated High-Conditioned Elliptic Function	100
	F_2	Rotated Bent Cigar Function	200
	F_3	Rotated Discus Function	300
Simple multimodal functions	F_4	Shifted and Rotated Rosenbrock's Function	400
	F_5	Shifted and Rotated Ackley's Function	500
	F_6	Shifted and Rotated Weierstrass Function	600
	F_7	Shifted and Rotated Griewank's Function	700
	F_8	Shifted Rastrigin's Function	800
	F_9	Shifted and Rotated Rastrigin's Function	900
	F_{10}	Shifted Schwefel's Function	1000
	F_{11}	Shifted and Rotated Schwefel's Function	1100
	F_{12}	Shifted and Rotated Katsuura Function	1200
	F_{13}	Shifted and Rotated HappyCat function	1300
	F_{14}	Shifted and Rotated HGBat Function	1400
	F_{15}	Shifted and Rotated Expanded Griewank's Plus Rosenbrock's Function	1500
	F_{16}	Shifted and Rotated Expanded Scaffer's F6 Function	1600
Hybrid functions	F_{17}	Hybrid function 1 ($N=3$)	1700
	F_{18}	Hybrid function 2 ($N=3$)	1800
	F_{19}	Hybrid function 3 ($N=4$)	1900
	F_{20}	Hybrid function 4 ($N=4$)	2000
	F_{21}	Hybrid function 5 ($N=5$)	2100
	F_{22}	Hybrid function 6 ($N=5$)	2200
Composition functions	F_{23}	Composition function 1 ($N=5$)	2300
	F_{24}	Composition function 2 ($N=3$)	2400
	F_{25}	Composition function 3 ($N=3$)	2500
	F_{26}	Composition function 4 ($N=5$)	2600
	F_{27}	Composition function 5 ($N=5$)	2700
	F_{28}	Composition function 6 ($N=5$)	2800
	F_{29}	Composition function 7 ($N=3$)	2900
	F_{30}	Composition function 8 ($N=3$)	3000
Search range: $[-100, 100]^D$			

Table 3: Parameter settings.

Algorithm	Parameter settings
BSA	$c_1 = c_2 = 1.5$, $a_1 = a_2 = 1$, FQ = 10
DE	$F = 0.5$, CR = 0.1
DMSDL-PSO	$c_1 = c_2 = 1.49445$, $F = 0.5$, CR = 0.1, sub-swarm = 10
DMSDL-BSA	$c_1 = c_2 = 1.5$, $a_1 = a_2 = 1$, FQ = 10, $F = 0.5$, CR = 0.1, sub-swarm = 10
DMSDL-QBSA	$c_1 = c_2 = 1.5$, $a_1 = a_2 = 1$, FQ = 10, $F = 0.5$, CR = 0.1, sub-swarm = 10

on any functions. DMSDL-BSA has got the minimum value of 0 on f_{11} and f_{13}. In summary, the DMSDL-QBSA has a superior global search capability on most multimodal functions when Dim = 2. Obviously, DMSDL-QBSA can find the best two parameters for RF that need to be optimized, because of its best global search capability.

In this section, it can be seen from Figures 1 and 2 and Tables 4–7 that DMSDL-QBSA can obtain the best function values for most cases. It indicates that the hybrid strategies of BSA, dynamic multi-swarm method, DE, and quantum behavior operators, lead to the bird moves towards the best solutions. And DMSDL-QBSA has well ability of searching for the best two parameters for RF with higher accuracy and efficiency.

2.3.4. Comparison on Benchmark Functions with Popular Algorithms. When comparing the timeliness and applicability of DMSDL-QBSA compared with several popular algorithms, such as GWO, WOA, SCA, GOA, and SSA, 18 benchmark functions are applied. And GWO, WOA, GOA and SSA are swarm intelligence algorithms. In this experiment, the dimension's size of these functions is10. The number of function evaluations (FEs) is100000. The maximum value (Max), the minimum value (Min), the mean value (Mean), and the variance (Var) obtained by several different algorithms are shown in Tables 8 and 9, where the best results are marked in bold.

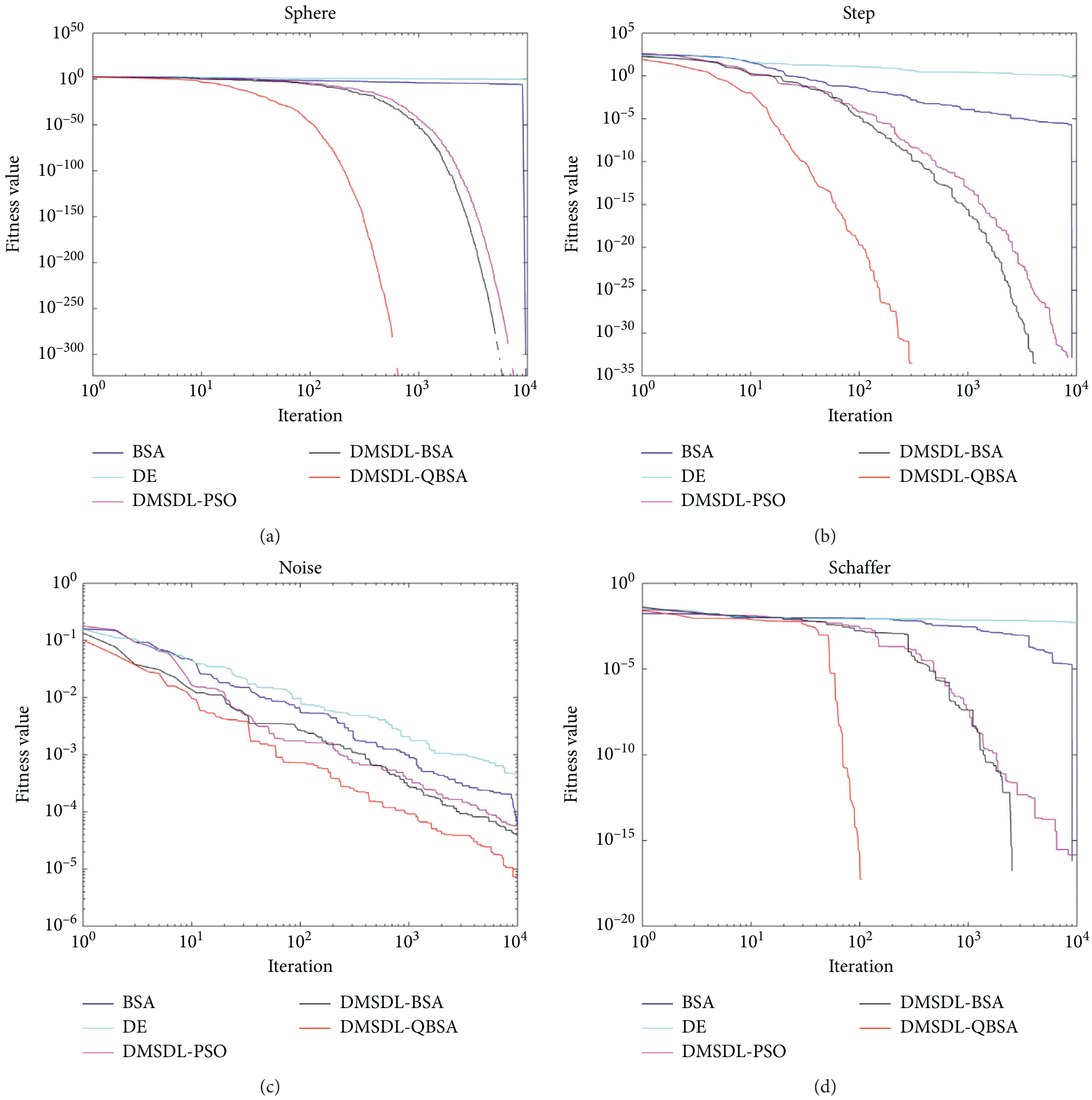


Figure 1: Fitness value curves of 5 hybrid algorithms on (a) f_1; (b) f_5; (c) f_6; and (d) f_9 (Dim = 2).

From the test results in Table 8, we can see that DMSDL-QBSA has the best performance on each unimodal function. GWO finds the value 0 on f_1, f_2, f_3, f_7, and f_8. WOA obtains 0 on f_1, f_2, and f_7. SSA works the best on f_1 and f_7. With the experiment of multimodal function evaluations, Table 9 shows that DMSDL-QBSA has the best performance on f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}, and f_{18}. SSA has the best performance on f_{10}. GWO gets the minimum on f_{11}. WOA and SCA obtains the optimal value on f_{11} and f_{13}. Obviously, compared with these popular algorithms, DMSDL-QBSA is a competitive algorithm for solving several functions and the swarm intelligence algorithms perform better than other algorithms. The results of Tables 8 and 9 show that DMSDL-QBSA has the best performance on the most test benchmark functions.

2.3.5. Comparison on CEC2014 Test Functions with Hybrid Algorithms. When comparing the comprehensive performance of proposed DMSDL-QBSA compared with several hybrid algorithms, such as BSA, DE, DMSDL-PSO, and DMSDL-BSA, 30 CEC2014 test functions are applied. In this experiment, the dimension's size (Dim) is set to 10. The number of function evaluations (FEs) is 100000. Experimental comparisons included the maximum value (Max), the minimum value (Min), the mean value (Mean), and the variance (Var) are given in Tables 10 and 11, where the best results are marked in bold.

Based on the mean value (Mean), on the CEC2014 test functions, DMSDL-QBSA has the best performance on F_2, F_3, F_4, F_6, F_7, F_8, F_9, F_{10}, F_{11}, F_{15}, F_{16}, F_{17}, F_{21}, F_{26}, F_{27}, F_{29}, and F_{30}. DMSDL-BSA does show an advantage on F_1, F_{12},

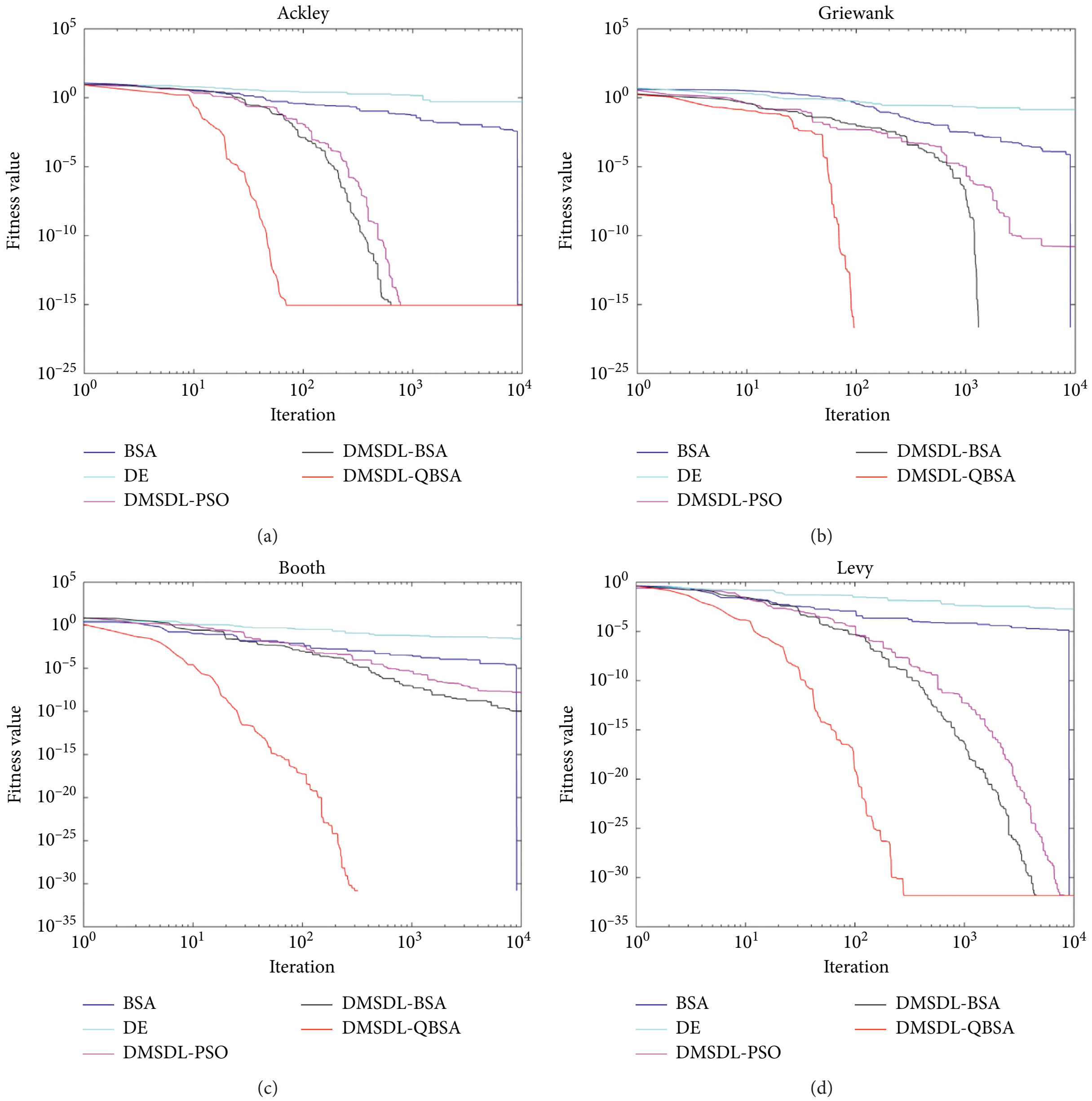


Figure 2: Fitness value curves of 5 hybrid algorithms on (a) f_{12}; (b) f_{13}; (c) f_{17}; and (d) f_{18} (Dim = 2).

F_{13}, F_{14}, F_{18}, F_{19}, F_{20}, F_{24}, F_{25}, and F_{28}. According to the results, we can observe that DMSDL-QBSA can find the minimal value on 17 CEC2014 test functions. DMSDL-BSA gets the minimum value on F_{1}, F_{12}, F_{13}, F_{14}, F_{18}, F_{19}, F_{24}, and F_{30}, and DMSDL-PSO obtains the minimum value on F_{4}, F_{7}, and F_{23}. Owing to enhance the capability of exploitation, DMSDL-QBSA is better than DMSDL-BSA and DMSDL-PSO on most functions. From the results of tests, it can be seen that DMSDL-QBSA performs better than BSA, DE, DMSDL-PSO, and DMSDL-BSA. It can be observed that DMSDL-QBSA obtains optimal value. It can be concluded that DMSDL-QBSA has better global search ability and better robustness on these test suites.

3. Optimize RF Classification Model Based on Improved BSA Algorithm

3.1. RF Classification Model. RF, as proposed by Breiman et al., is an ensemble learning model based on bagging and random subspace methods. The whole modeling process includes building decision trees and decision processes. The process of constructing decision trees is mainly composed of

Table 4: Comparison on nine unimodal functions with 5 hybrid algorithms (Dim = 10).

Function	Term	BSA	DE	DMSDL-PSO	DMSDL-BSA	DMSDL-QBSA
f_1	Max	$1.0317E+04$	$1.2466E+04$	$1.7232E+04$	$1.0874E+04$	$\mathbf{7.4436E+01}$
	Min	$\mathbf{0}$	$4.0214E+03$	$1.6580E-02$	$\mathbf{0}$	$\mathbf{0}$
	Mean	$6.6202E+00$	$4.6622E+03$	$6.0643E+01$	$6.3444E+00$	$\mathbf{3.3920E-02}$
	Var	$2.0892E+02$	$8.2293E+02$	$7.3868E+02$	$1.9784E+02$	$\mathbf{1.2343E+00}$
f_2	Max	$3.3278E+02$	$1.8153E+02$	$8.0436E+01$	$4.9554E+02$	$\mathbf{2.9845E+01}$
	Min	$4.9286E-182$	$1.5889E+01$	$1.0536E-01$	$3.0074E-182$	$\mathbf{0}$
	Mean	$5.7340E-02$	$1.8349E+01$	$2.9779E+00$	$6.9700E-02$	$\mathbf{1.1220E-02}$
	Var	$3.5768E+00$	$4.8296E+00$	$2.4966E+00$	$5.1243E+00$	$\mathbf{4.2864E-01}$
f_3	Max	$1.3078E+04$	$1.3949E+04$	$1.9382E+04$	$1.2899E+04$	$\mathbf{8.4935E+01}$
	Min	$3.4873E-250$	$4.0327E+03$	$6.5860E-02$	$1.6352E-249$	$\mathbf{0}$
	Mean	$7.6735E+00$	$4.6130E+03$	$8.6149E+01$	$7.5623E+00$	$\mathbf{3.3260E-02}$
	Var	$2.4929E+02$	$8.4876E+02$	$8.1698E+02$	$2.4169E+02$	$\mathbf{1.2827E+00}$
f_4	Max	$2.5311E+09$	$2.3900E+09$	$4.8639E+09$	$3.7041E+09$	$\mathbf{3.5739E+08}$
	Min	$5.2310E+00$	$2.7690E+08$	$8.4802E+00$	$\mathbf{5.0021E+00}$	$8.9799E+00$
	Mean	$6.9192E+05$	$3.3334E+08$	$1.1841E+07$	$9.9162E+05$	$\mathbf{6.8518E+04}$
	Var	$3.6005E+07$	$1.4428E+08$	$1.8149E+08$	$4.5261E+07$	$\mathbf{4.0563E+06}$
f_5	Max	$1.1619E+04$	$1.3773E+04$	$1.6188E+04$	$1.3194E+04$	$\mathbf{5.1960E+03}$
	Min	$5.5043E-15$	$5.6109E+03$	$1.1894E-02$	$\mathbf{4.2090E-15}$	$1.5157E+00$
	Mean	$5.9547E+00$	$6.3278E+03$	$5.2064E+01$	$6.5198E+00$	$\mathbf{3.5440E+00}$
	Var	$2.0533E+02$	$9.6605E+02$	$6.2095E+02$	$2.2457E+02$	$\mathbf{7.8983E+01}$
f_6	Max	$3.2456E+00$	$7.3566E+00$	$8.9320E+00$	$2.8822E+00$	$\mathbf{1.4244E+00}$
	Min	$1.3994E-04$	$1.2186E+00$	$2.2482E-03$	$8.2911E-05$	$\mathbf{1.0911E-05}$
	Mean	$2.1509E-03$	$1.4021E+00$	$1.1982E-01$	$1.9200E-03$	$\mathbf{6.1476E-04}$
	Var	$5.3780E-02$	$3.8482E-01$	$3.5554E-01$	$5.0940E-02$	$\mathbf{1.9880E-02}$
f_7	Max	$4.7215E+02$	$6.7534E+02$	$5.6753E+02$	$5.3090E+02$	$\mathbf{2.3468E+02}$
	Min	$\mathbf{0}$	$2.2001E+02$	$5.6300E-02$	$\mathbf{0}$	$\mathbf{0}$
	Mean	$2.4908E-01$	$2.3377E+02$	$9.2909E+00$	$3.0558E-01$	$\mathbf{9.4500E-02}$
	Var	$8.5433E+00$	$3.3856E+01$	$2.2424E+01$	$1.0089E+01$	$\mathbf{3.5569E+00}$
f_8	Max	$3.2500E+02$	$2.4690E+02$	$2.7226E+02$	$2.8001E+02$	$\mathbf{1.7249E+02}$
	Min	$1.4678E-239$	$8.3483E+01$	$5.9820E-02$	$8.9624E-239$	$\mathbf{0}$
	Mean	$1.9072E-01$	$9.1050E+01$	$7.9923E+00$	$2.3232E-01$	$\mathbf{8.1580E-02}$
	Var	$6.3211E+00$	$1.3811E+01$	$1.7349E+01$	$6.4400E+00$	$\mathbf{2.9531E+00}$

n_{tree} decision trees, and each of which consists of nonleaf nodes and leaf nodes. The leaf node is a child node of the node branch. It is supposed that the dataset has M attributes. When each leaf node of the decision tree needs to be segmented, the m_{try} attributes are randomly selected from the M attributes as the reselected splitting variables of this node. This process can be defined as follows:

$$S_j = \left\{ P_i \,\middle|\, i = 1, 2, \ldots, m_{\text{try}} \right\}, \tag{18}$$

where S_j is the splitting variable of the j-th leaf node of the decision tree, and P_i is the probability that m_{try} reselected attributes are selected as the splitting attribute of the node.

The nonleaf node is a parent node that classifies training data as a left or right child node. The function of k-th decision tree is as follows:

$$h_k(c \mid x) = \begin{cases} 0, & f_l(x_k) < f_l(x_k) + \tau, \\ 1, & f_l(x_k) \geq f_l(x_k) + \tau, \end{cases} \quad l = 1, 2, \ldots, n_{\text{tree}}, \tag{19}$$

where $c = \{0 \text{ or } 1\}$, where the symbol 0 indicates that the k-th row of data is classified as a negative label and the symbol 1 indicates that the k-th row of data is classified as a positive label. Here, f_l is the training function of the l-th decision tree based on the splitting variable S. X_k is the k-th row of data in the dataset by random sampling with replacement. The symbol τ is a positive constant, which is used as the threshold value of the training decision.

When decision processes are trained, each row of data will be input into a leaf node of each decision tree. The average of n_{tree} decision tree classification results is used as the final classification result. This process can be written mathematically as follows:

$$c_k = l \times \frac{1}{n_{\text{tree}}} \sum_{k=1}^{n_{\text{tree}}} h_k(c \mid x), \tag{20}$$

where l is the number of decision trees which judged k-th row of data as c.

From the above principle, we can see that it is mainly necessary to determine two parameters of n_{tree} and m_{try} in the RF modeling process. In order to verify the influence of these two parameters on the classification accuracy of the RF classification model, the Ionosphere dataset is used to test the influence of the two parameters on the performance of

Table 5: Comparison on nine unimodal functions with hybrid algorithms (Dim = 2).

Function	Term	BSA	DE	DMSDL-PSO	DMSDL-BSA	DMSDL-QBSA
f_1	Max	1.8411E + 02	3.4713E + 02	2.9347E + 02	1.6918E + 02	**1.6789E + 00**
	Min	**0**	5.7879E − 01	**0**	**0**	3.2988E − 238
	Mean	5.5890E − 02	1.1150E + 00	1.6095E − 01	3.6580E − 02	**4.2590E − 03**
	Var	2.6628E + 00	5.7218E + 00	5.4561E + 00	2.0867E + 00	**7.5858E − 02**
f_2	Max	2.2980E + 00	2.1935E + 00	3.2363E + 00	3.1492E + 00	**1.1020E + 00**
	Min	5.5923E − 266	8.2690E − 02	2.9096E − 242	3.4367E − 241	**0**
	Mean	9.2769E − 04	9.3960E − 02	7.4900E − 03	1.2045E − 03	**2.1565E − 04**
	Var	3.3310E − 02	6.9080E − 02	4.0100E − 02	4.2190E − 02	**1.3130E − 02**
f_3	Max	**1.0647E + 02**	1.3245E + 02	3.6203E + 02	2.3793E + 02	1.3089E + 02
	Min	**0**	5.9950E − 01	**0**	**0**	**0**
	Mean	2.3040E − 02	8.5959E − 01	2.5020E − 01	6.3560E − 02	**1.7170E − 02**
	Var	**1.2892E + 00**	2.8747E + 00	7.5569E + 00	2.9203E + 00	1.3518E + 00
f_4	Max	1.7097E + 02	6.1375E + 01	6.8210E + 01	5.9141E + 01	**1.4726E + 01**
	Min	1.6325E − 21	4.0940E − 02	8.2726E − 13	3.4830E − 25	**0**
	Mean	2.2480E − 02	9.3940E − 02	1.5730E − 02	**1.1020E − 02**	1.4308E − 02
	Var	1.7987E + 00	7.4859E − 01	7.6015E − 01	6.4984E − 01	**2.7598E − 01**
f_5	Max	1.5719E + 02	2.2513E + 02	3.3938E + 02	1.8946E + 02	**8.7078E + 01**
	Min	**0**	7.0367E − 01	**0**	**0**	**0**
	Mean	3.4380E − 02	1.8850E + 00	1.7082E − 01	5.0090E − 02	**1.1880E − 02**
	Var	1.9018E + 00	5.6163E + 00	5.9868E + 00	2.4994E + 00	**9.1749E − 01**
f_6	Max	1.5887E − 01	1.5649E − 01	1.5919E − 01	1.3461E − 01	**1.0139E − 01**
	Min	2.5412E − 05	4.5060E − 04	5.9140E − 05	4.1588E − 05	**7.3524E − 06**
	Mean	2.3437E − 04	1.3328E − 03	6.0989E − 04	2.3462E − 04	**9.2394E − 05**
	Var	2.4301E − 03	3.6700E − 03	3.5200E − 03	1.9117E − 03	**1.4664E − 03**
f_7	Max	3.5804E + 00	2.8236E + 00	**1.7372E + 00**	2.7513E + 00	1.9411E + 00
	Min	**0**	7.6633E − 03	**0**	**0**	**0**
	Mean	8.5474E − 04	1.6590E − 02	8.6701E − 04	7.6781E − 04	**3.1439E − 04**
	Var	4.4630E − 02	6.0390E − 02	2.4090E − 02	3.7520E − 02	**2.2333E − 02**
f_8	Max	4.3247E + 00	2.1924E + 00	5.3555E + 00	3.3944E + 00	**5.5079E − 01**
	Min	**0**	8.6132E − 03	**0**	**0**	**0**
	Mean	1.1649E − 03	1.9330E − 02	1.7145E − 03	7.3418E − 04	**6.9138E − 05**
	Var	5.9280E − 02	4.7800E − 02	7.5810E − 02	4.1414E − 02	**5.7489E − 03**
f_9	Max	2.7030E − 02	3.5200E − 02	1.7240E − 02	4.0480E − 02	**2.5230E − 02**
	Min	**0**	5.0732E − 03	**0**	**0**	**0**
	Mean	6.1701E − 05	6.2500E − 03	8.8947E − 04	8.4870E − 05	**2.7362E − 05**
	Var	7.6990E − 04	1.3062E − 03	2.0400E − 03	9.6160E − 04	**5.5610E − 04**

the RF model, as shown in Figure 3, where the horizontal axis represents n_{tree} and m_{try}, respectively, and the vertical axis represents the accuracy of the RF classification model.

(1) Parameter analysis of n_{tree}

When the number of predictor variables m_{try} is set to 6, the number of decision trees n_{tree} is cyclically set from 0 to 1000 at intervals of 20. And the evolutionary progress of RF classification model accuracy with the change of n_{tree} is shown in Figure 3(a). From the curve in Figure 3(a), we can see that the accuracy of RF is gradually improved with the increase of the number N of decision trees. However, when the number of decision trees n_{tree} is greater than a certain value, the improvement of RF performance has become gentle without obvious improvement, but the running time becomes longer.

(2) Parameter analysis of m_{try}

When the number of decision trees n_{tree} is set to 500, the number of predictor variables m_{try} is cyclically set from 1 to 32. The limit of m_{try} is set to 32, because the number of attributes of the Ionosphere dataset is 32. And the obtained curve of RF classification model accuracy with m_{try} transformation is shown in Figure 3(b). And we can see that with the increase of the splitting property of the selection, the classification performance of RF is gradually improved, but when the number of predictor variables m_{try} is greater than 9, the RF generates overfitting and the accuracy of RF begins to decrease. The main reason is that too many split attributes are selected, which resulted in the same splitting attributes which are owned by a large number of decision trees. This reduced the diversity of decision trees.

Table 6: Comparison on nine multimodal functions with hybrid algorithms (Dim = 10).

Function	Term	BSA	DE	DMSDL-PSO	DMSDL-BSA	DMSDL-QBSA
f_{10}	Max	2.8498E + 03	2.8226E + 03	3.0564E + 03	**2.7739E + 03**	2.8795E + 03
	Min	**1.2553E + 02**	1.8214E + 03	1.2922E + 03	1.6446E + 02	1.1634E + 03
	Mean	**2.5861E + 02**	1.9229E + 03	1.3185E + 03	3.1119E + 02	1.2729E + 03
	Var	2.4093E + 02	**1.2066E + 02**	1.2663E + 02	2.5060E + 02	1.2998E + 02
f_{11}	Max	1.2550E + 02	1.0899E + 02	1.1806E + 02	1.1243E + 02	**9.1376E + 01**
	Min	**0**	6.3502E + 01	1.0751E + 01	**0**	**0**
	Mean	2.0417E − 01	6.7394E + 01	3.9864E + 01	1.3732E − 01	**6.8060E − 02**
	Var	3.5886E + 00	5.8621E + 00	1.3570E + 01	3.0325E + 00	**2.0567E + 00**
f_{12}	Max	2.0021E + 01	1.9910E + 01	1.9748E + 01	1.9254E + 01	**1.8118E + 01**
	Min	**8.8818E − 16**	1.6575E + 01	7.1700E − 02	**8.8818E − 16**	**8.8818E − 16**
	Mean	3.0500E − 02	1.7157E + 01	3.0367E + 00	3.8520E − 02	**1.3420E − 02**
	Var	5.8820E − 01	5.2968E − 01	1.6585E + 00	6.4822E − 01	**4.2888E − 01**
f_{13}	Max	1.0431E + 02	1.3266E + 02	1.5115E + 02	1.2017E + 02	**6.1996E + 01**
	Min	**0**	4.5742E + 01	2.1198E − 01	**0**	**0**
	Mean	6.1050E − 02	5.2056E + 01	3.0613E + 00	6.9340E − 02	**2.9700E − 02**
	Var	1.8258E + 00	8.3141E + 00	1.5058E + 01	2.2452E + 00	**1.0425E + 00**
f_{14}	Max	**8.4576E + 06**	3.0442E + 07	5.3508E + 07	6.2509E + 07	8.5231E + 06
	Min	1.7658E − 13	1.9816E + 06	4.5685E − 05	**1.6961E − 13**	5.1104E − 01
	Mean	1.3266E + 03	3.1857E + 06	6.8165E + 04	8.8667E + 03	**1.1326E + 03**
	Var	9.7405E + 04	1.4876E + 06	1.4622E + 06	6.4328E + 05	**8.7645E + 04**
f_{15}	Max	1.8310E + 08	1.4389E + 08	1.8502E + 08	1.4578E + 08	**2.5680E + 07**
	Min	1.7942E − 11	1.0497E + 07	2.4500E − 03	**1.1248E − 11**	9.9870E − 01
	Mean	3.7089E + 04	1.5974E + 07	2.0226E + 05	3.8852E + 04	**3.5739E + 03**
	Var	2.0633E + 06	1.0724E + 07	4.6539E + 06	2.1133E + 06	**2.6488E + 05**
f_{16}	Max	1.3876E + 01	1.4988E + 01	1.4849E + 01	1.3506E + 01	**9.3280E + 00**
	Min	4.2410E − 174	6.8743E + 00	2.5133E − 02	7.3524E − 176	**0**
	Mean	1.3633E − 02	7.2408E + 00	2.5045E + 00	1.3900E − 02	**5.1800E − 03**
	Var	3.3567E − 01	7.7774E − 01	1.0219E + 00	3.4678E − 01	**1.7952E − 01**
f_{18}	Max	3.6704E + 01	3.6950E + 01	2.8458E + 01	2.6869E + 01	**2.4435E + 01**
	Min	2.0914E − 11	9.7737E + 00	3.3997E − 03	**5.9165E − 12**	7.5806E − 01
	Mean	6.5733E − 02	1.2351E + 01	6.7478E − 01	**5.6520E − 02**	7.9392E − 01
	Var	6.7543E − 01	2.8057E + 00	1.4666E + 00	6.4874E − 01	**3.6928E − 01**

In summary, for the RF classification model to obtain the ideal optimal solution, the selection of the number of decision trees n_{tree} and the number of predictor variables m_{try} are very important. And the classification accuracy of the RF classification model can only be optimized by the comprehensive optimization of these two parameters. So, it is necessary to use the proposed algorithm to find a suitable set of RF parameters. Next, we will optimize the RF classification model by the improved BSA proposed in Section 2.

3.2. RF Model Based on an Improved Bird Swarm Algorithm. Improved bird swarm algorithm optimized RF classification model (DMSDL-QBSA-RF) is based on the improved bird swarm algorithm optimized the RF classification model and introduced the training dataset into the training process of the RF classification model, finally getting the DMSDL-QBSA-RF classification model. The main idea is to construct a two-dimensional fitness function containing RF's two parameters n_{tree} and m_{try} as the optimization target of DMSDL-QBSA, so as to obtain a set of grouping parameters and make the RF classification model obtain the best classification accuracy. The specific algorithm steps are shown as in Algorithm 2.

3.3. Simulation Experiment and Analysis. In order to test the performance of the improved DMSDL-QBSA-RF classification model, we compare the improved classification model with the standard RF model, BSA-RF model, and DMSDL-BSA-RF model on 8 two-dimensional UCI datasets. The DMSDL-BSA-RF classification model is an RF classification model optimized by BSA without quantum behavior. In our experiment, each of datasets is divided into two parts: 70% of the dataset is as training set and the remaining 30% is as a test set. The average classification accuracies of 10 independent runs of each model are recorded in Table 12, where the best results are marked in bold.

From the accuracy results in Table 12, we can see that the DMSDL-QBSA-RF classification model can get best accuracy on each UCI dataset except magic dataset. And the DMSDL-BSA-RF classification model has got best accuracy on magic dataset. Then, compared with the standard RF model, the accuracy of the DMSDL-QBSA-RF classification model can get better accuracy which is increased by about 10%. Finally, the

Table 7: Comparison on nine multimodal functions with hybrid algorithms (Dim = 2).

Function	Term	BSA	DE	DMSDL-PSO	DMSDL-BSA	DMSDL-QBSA
f_{10}	Max	**2.0101E+02**	2.8292E+02	2.9899E+02	2.4244E+02	2.8533E+02
	Min	**2.5455E−05**	3.7717E+00	2.3690E+01	**2.5455E−05**	9.4751E+01
	Mean	**3.4882E−01**	8.0980E+00	2.5222E+01	4.2346E−01	9.4816E+01
	Var	5.7922E+00	1.0853E+01	1.5533E+01	5.6138E+00	**2.8160E+00**
f_{11}	Max	**4.7662E+00**	4.7784E+00	1.1067E+01	8.7792E+00	8.1665E+00
	Min	**0**	3.8174E−01	**0**	**0**	**0**
	Mean	**2.7200E−03**	6.0050E−01	3.1540E−02	4.2587E−03	3.7800E−03
	Var	**8.6860E−02**	3.1980E−01	2.4862E−01	1.2032E−01	1.3420E−01
f_{12}	Max	9.6893E+00	**8.1811E+00**	1.1635E+01	9.1576E+00	8.4720E+00
	Min	**8.8818E−16**	5.1646E−01	**8.8818E−16**	**8.8818E−16**	**8.8818E−16**
	Mean	**9.5600E−03**	6.9734E−01	3.4540E−02	9.9600E−03	2.8548E−03
	Var	2.1936E−01	6.1050E−01	2.5816E−01	2.1556E−01	**1.1804E−01**
f_{13}	Max	4.4609E+00	4.9215E+00	4.1160E+00	1.9020E+00	**1.6875E+00**
	Min	**0**	1.3718E−01	**0**	**0**	**0**
	Mean	1.9200E−03	1.7032E−01	1.8240E−02	1.4800E−03	**5.7618E−04**
	Var	6.6900E−02	1.3032E−01	1.8202E−01	3.3900E−02	**2.2360E−02**
f_{14}	Max	1.0045E+01	1.9266E+03	1.9212E+01	5.7939E+02	**8.2650E+00**
	Min	**2.3558E−31**	1.3188E−01	**2.3558E−31**	**2.3558E−31**	**2.3558E−31**
	Mean	4.1600E−03	3.5402E−01	1.0840E−02	6.1420E−02	**1.3924E−03**
	Var	1.7174E−01	1.9427E+01	3.9528E−01	5.8445E+00	**8.7160E−02**
f_{15}	Max	6.5797E+04	4.4041E+03	1.4412E+05	8.6107E+03	**2.6372E+00**
	Min	**1.3498E−31**	9.1580E−02	**1.3498E−31**	**1.3498E−31**	7.7800E−03
	Mean	7.1736E+00	8.9370E−01	1.7440E+01	9.0066E−01	**8.2551E−03**
	Var	6.7678E+02	5.4800E+01	1.4742E+03	8.7683E+01	**2.8820E−02**
f_{16}	Max	6.2468E−01	6.4488E−01	5.1564E−01	8.4452E−01	**3.9560E−01**
	Min	6.9981E−08	2.5000E−03	1.5518E−240	2.7655E−07	**0**
	Mean	2.7062E−04	6.9400E−03	6.8555E−04	2.0497E−04	**6.1996E−05**
	Var	1.0380E−02	1.7520E−02	8.4600E−03	1.0140E−02	**4.4000E−03**
f_{17}	Max	5.1946E+00	3.6014E+00	2.3463E+00	6.9106E+00	**1.2521E+00**
	Min	2.6445E−11	2.6739E−02	**0**	1.0855E−10	**0**
	Mean	1.9343E−03	5.1800E−02	1.2245E−03	2.8193E−03	**1.5138E−04**
	Var	7.3540E−02	1.2590E−01	4.1620E−02	1.1506E−01	**1.2699E−02**
f_{18}	Max	5.0214E−01	**3.4034E−01**	4.1400E−01	3.7422E−01	4.0295E−01
	Min	**1.4998E−32**	1.9167E−03	**1.4998E−32**	**1.4998E−32**	**1.4998E−32**
	Mean	1.0967E−04	4.1000E−03	1.8998E−04	1.4147E−04	**6.0718E−05**
	Var	6.1800E−03	1.0500E−02	6.5200E−03	5.7200E−03	**4.4014E−03**

DMSDL-QBSA-RF classification model has got the best accuracy on appendicitis dataset which is up to 93.55%. In summary, the DMSDL-QBSA-RF classification model has validity on most datasets and a good performance on them.

4. Oil Layer Classification Application

4.1. Design of Oil Layer Classification System. The block diagram of the oil layer classification system based on the improved DMSDL-QBSA-RF is shown in Figure 4. The oil layer classification can be simplified by the following five steps:

Step 1. The selection of the actual logging datasets is intact and full-scale. At the same time, the datasets should be closely related to rock sample analysis. The dataset should be relatively independent. The dataset is randomly divided into two parts of training and testing samples.

Step 2. In order to better understand the relationship between independent variables and dependent variables and reduce the sample information attribute, the dataset continuous attribute should be discretized by using a greedy algorithm.

Step 3. In order to improve the calculation speed and classification accuracy, we use the covering rough set method [43] to realize the attribute reduction. After attribute reduction, normalization of the actual logging datasets is carried out to avoid computational saturation.

Step 4. In the DMSDL-QBSA-RF layer classification model, we input the actual logging dataset after attribute reduction, use a DMSDL-QBSA-RF layer classification algorithm to train, and finally get the DMSDL-QBSA-RF layer classification model.

Table 8: Comparison on 8 unimodal functions with popular algorithms (Dim = 10, FEs = 100000).

Function	Term	GWO	WOA	SCA	GOA	SSA	DMSDL-QBSA
f_1	Max	1.3396E + 04	1.4767E + 04	1.3310E + 04	2.0099E + 01	4.8745E + 03	**9.8570E − 01**
	Min	**0**	**0**	4.2905E − 293	8.6468E − 17	**0**	**0**
	Mean	3.5990E + 00	4.7621E + 00	1.4014E + 02	7.0100E − 02	4.5864E + 00	**1.0483E − 04**
	Var	1.7645E + 02	2.0419E + 02	8.5054E + 02	4.4200E − 01	1.4148E + 02	**9.8725E − 03**
f_2	Max	3.6021E + 02	2.5789E + 03	6.5027E + 01	9.3479E + 01	3.4359E + 01	**3.2313E − 01**
	Min	**0**	**0**	9.8354E − 192	2.8954E − 03	2.4642E − 181	**0**
	Mean	5.0667E − 02	2.9480E − 01	4.0760E − 01	3.1406E + 00	1.7000E − 02	**1.1278E − 04**
	Var	3.7270E + 00	2.6091E + 01	2.2746E + 00	3.9264E + 00	5.4370E − 01	**4.8000E − 03**
f_3	Max	1.8041E + 04	1.6789E + 04	2.4921E + 04	6.5697E + 03	1.1382E + 04	**4.0855E + 01**
	Min	**0**	1.0581E − 18	7.6116E − 133	2.8796E − 01	3.2956E − 253	1.5918E − 264
	Mean	7.4511E + 00	2.8838E + 02	4.0693E + 02	4.8472E + 02	9.2062E + 00	**4.8381E − 03**
	Var	2.6124E + 02	1.4642E + 03	1.5913E + 03	7.1786E + 02	2.9107E + 02	**4.1383E − 01**
f_4	Max	2.1812E + 09	5.4706E + 09	8.4019E + 09	1.1942E + 09	4.9386E + 08	**7.5188E + 01**
	Min	4.9125E + 00	3.5695E + 00	5.9559E + 00	2.2249E + 02	4.9806E + 00	**2.9279E − 13**
	Mean	4.9592E + 05	2.4802E + 06	4.4489E + 07	5.0021E + 06	3.3374E + 05	**2.4033E − 02**
	Var	2.9484E + 07	1.1616E + 08	4.5682E + 08	3.9698E + 07	1.1952E + 07	**9.7253E − 01**
f_5	Max	1.8222E + 04	1.5374E + 04	1.5874E + 04	1.2132E + 03	1.6361E + 04	**1.8007E + 00**
	Min	1.1334E − 08	8.3228E − 09	2.3971E − 01	2.7566E − 10	2.6159E − 16	**1.0272E − 33**
	Mean	5.1332E + 00	5.9967E + 00	1.2620E + 02	5.8321E + 01	8.8985E + 00	**2.3963E − 04**
	Var	2.3617E + 02	2.3285E + 02	8.8155E + 02	1.0872E + 02	2.9986E + 02	**1.8500E − 02**
f_6	Max	7.4088E + 00	8.3047E + 00	8.8101E + 00	6.8900E − 01	4.4298E + 00	**2.5787E − 01**
	Min	1.8112E − 05	3.9349E − 05	4.8350E − 05	8.9528E − 02	4.0807E − 05	1.0734E − 04
	Mean	1.8333E − 03	3.2667E − 03	4.8400E − 02	9.3300E − 02	2.1000E − 03	**5.2825E − 04**
	Var	9.4267E − 02	1.0077E − 01	2.9410E − 01	3.5900E − 02	6.5500E − 02	**4.6000E − 03**
f_7	Max	7.3626E + 02	6.8488E + 02	8.0796E + 02	3.9241E + 02	8.2036E + 02	**1.4770E + 01**
	Min	**0**	**0**	1.9441E − 292	7.9956E − 07	**0**	**0**
	Mean	2.0490E − 01	2.8060E − 01	4.9889E + 00	1.6572E + 01	2.7290E − 01	**1.8081E − 03**
	Var	9.5155E + 00	1.0152E + 01	3.5531E + 01	2.3058E + 01	1.0581E + 01	**1.6033E − 01**
f_8	Max	1.2749E + 03	5.9740E + 02	3.2527E + 02	2.3425E + 02	2.0300E + 02	**2.0423E + 01**
	Min	**0**	4.3596E − 35	1.5241E − 160	3.6588E − 05	1.0239E − 244	**0**
	Mean	3.1317E − 01	1.0582E + 01	1.0457E + 01	1.2497E + 01	2.1870E − 01	**2.6947E − 03**
	Var	1.4416E + 01	4.3485E + 01	3.5021E + 01	2.5766E + 01	6.2362E + 00	**2.1290E − 01**

Step 5. The whole oil section is identified by the trained DMSDL-QBSA-RF layer classification model, and we output the classification results.

In order to verify the application effect of the DMSDL-QBSA-RF layer classification model, we select three actual logging datasets of oil and gas wells to train and test.

4.2. Practical Application.

In Section 2.3, the performance of the proposed DMSDL-QBSA is simulated and analyzed on benchmark functions. And in Section 3.3, the effectiveness of the improved RF classification model optimized by the proposed DMSDL-QBSA is tested and verified on two-dimensional UCI datasets. In order to test the application effect of the improved DMSDL-QBSA-RF layer classification model, three actual logging datasets are adopted and recorded as mathematical problems in engineering $W1$, $W2$, and $W3$. The $W1$ is a gas well in Xian (China), the W2 is a gas well in Shanxi (China), and the $W3$ is an oil well in Xinjiang (China). The depth and the corresponding rock sample analysis samples of the three wells selected in the experiment are as shown in Table 13.

Attribute reduction on the actual logging datasets is performed before the training of the DMSDL-QBSA-RF classification model on the training dataset, as shown in Table 14. Then, these attributes are normalized as shown in Figure 5, where the horizontal axis represents the depth and the vertical axis represents the normalized value.

The logging dataset after attribute reduction and normalization is used to train the oil and gas layer classification model. In order to measure the performance of the DMSDL-QBSA-RF classification model, we compare the improved classification model with several popular oil and gas layer classification models. These classification models are the standard RF model, SVM model, BSA-RF model, and DMSDL-BSA-RF model. Here, the RF classification model was first applied to the field of logging. In order to evaluate the performance of the recognition model, we select the following performance indicators:

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(f_i - y_i\right)^2}, \\ \text{MAE} &= \frac{1}{N}\sum_{i=1}^{N}\left|f_i - y_i\right|, \end{aligned} \tag{21}$$

Table 9: Comparison on 8 multimodal functions with popular algorithms (Dim = 10, FEs = 100000).

Function	Term	GWO	WOA	SCA	GOA	SSA	DMSDL-QBSA
f_{10}	Max	2.9544E + 03	2.9903E + 03	2.7629E + 03	2.9445E + 03	3.2180E + 03	3.0032E + 03
	Min	1.3037E + 03	8.6892E − 04	1.5713E + 03	1.3438E + 03	**1.2839E − 04**	1.2922E + 03
	Mean	1.6053E + 03	1.4339E + 01	1.7860E + 03	1.8562E + 03	**2.5055E + 02**	1.2960E + 03
	Var	2.6594E + 02	1.2243E + 02	1.6564E + 02	5.1605E + 02	3.3099E + 02	**5.8691E + 01**
f_{11}	Max	1.3792E + 02	1.2293E + 02	1.2313E + 02	1.1249E + 02	3.2180E + 03	**1.8455E + 01**
	Min	**0**	**0**	**0**	1.2437E + 01	1.2839E − 04	**0**
	Mean	4.4220E − 01	1.1252E + 00	9.9316E + 00	2.8378E + 01	2.5055E + 02	**3.2676E − 03**
	Var	4.3784E + 00	6.5162E + 00	1.9180E + 01	1.6240E + 01	3.3099E + 02	**1.9867E − 01**
f_{12}	Max	2.0257E + 01	2.0043E + 01	1.9440E + 01	1.6623E + 01	3.2180E + 03	**1.9113E + 00**
	Min	4.4409E − 15	3.2567E − 15	3.2567E − 15	2.3168E + 00	1.2839E − 04	**8.8818E − 16**
	Mean	1.7200E − 02	4.2200E − 02	8.8870E − 01	5.5339E + 00	2.5055E + 02	**3.1275E − 04**
	Var	4.7080E − 01	6.4937E − 01	3.0887E + 00	2.8866E + 00	3.3099E + 02	**2.2433E − 02**
f_{13}	Max	1.5246E + 02	1.6106E + 02	1.1187E + 02	6.1505E + 01	3.2180E + 03	**3.1560E − 01**
	Min	3.3000E − 03	**0**	**0**	2.4147E − 01	1.2839E − 04	**0**
	Mean	4.6733E − 02	9.3867E − 02	1.2094E + 00	3.7540E + 00	2.5055E + 02	**3.3660E − 05**
	Var	1.9297E + 00	2.6570E + 00	6.8476E + 00	4.1936E + 00	3.3099E + 02	**3.1721E − 03**
f_{14}	Max	9.5993E + 07	9.9026E + 07	5.9355E + 07	6.1674 E + 06	3.2180E + 03	**6.4903E − 01**
	Min	3.8394E − 09	1.1749E − 08	9.6787E − 03	1.8099E − 04	1.2839E − 04	**4.7116E − 32**
	Mean	1.2033E + 04	3.5007E + 04	4.8303E + 05	1.0465E + 04	2.5055E + 02	**8.9321E − 05**
	Var	9.8272E + 05	1.5889E + 06	4.0068E + 06	1.9887E + 05	3.3099E + 02	**6.8667E − 03**
f_{15}	Max	2.2691E + 08	2.4717E + 08	1.1346E + 08	2.8101E + 07	3.2180E + 03	**1.6407E − 01**
	Min	3.2467E − 02	4.5345E − 08	1.1922E − 01	3.5465E − 05	1.2839E − 04	**1.3498E − 32**
	Mean	2.9011E + 04	4.3873E + 04	6.5529E + 05	7.2504E + 04	2.5055E + 02	**6.7357E − 05**
	Var	2.3526E + 06	2.7453E + 06	7.1864E + 06	1.2814E + 06	3.3099E + 02	**2.4333E − 03**
f_{16}	Max	1.7692E + 01	1.7142E + 01	1.6087E + 01	8.7570E + 00	3.2180E + 03	**1.0959E + 00**
	Min	2.6210E − 07	0.0000E + 00	6.2663E − 155	1.0497E − 02	1.2839E − 04	**0**
	Mean	1.0133E − 02	3.9073E − 01	5.9003E − 01	2.4770E + 00	2.5055E + 02	**1.5200E − 04**
	Var	3.0110E − 01	9.6267E − 01	1.4701E + 00	1.9985E + 00	3.3099E + 02	**1.1633E − 02**
f_{18}	Max	4.4776E + 01	4.3588E + 01	3.9095E + 01	1.7041E + 01	3.2180E + 03	**6.5613E − 01**
	Min	1.9360E − 01	9.4058E − 08	2.2666E − 01	3.9111E + 00	1.2839E − 04	**1.4998E − 32**
	Mean	2.1563E − 01	4.7800E − 02	9.8357E − 01	5.4021E + 00	2.5055E + 02	**9.4518E − 05**
	Var	5.8130E − 01	1.0434E + 00	3.0643E + 00	1.6674E + 00	3.3099E + 02	**7.0251E − 03**

where y_i and f_i are the classification output value and the expected output value, respectively.

RMSE is used to evaluate the accuracy of each classification model. MAE is used to show actual forecasting errors. Table 15 records the performance indicator data of each classification model, and the best results are marked in bold. The smaller the RMSE and MAE, the better the classification model performs.

From the performance indicator data of each classification model in Table 15, we can see that the DMSDL-QBSA-RF classification model can get the best recognition accuracy and all the accuracies are up to 90%. The recognition accuracy of the proposed classification model for W3 is up to 99.73%, and it has superior performance for oil and gas layer classification in other performance indicators and different wells. Secondly, DMSDL-QBSA can improve the performance of RF, and the parameters found by DMSDL-QBSA used in the RF classification model can improve the classification accuracy and keep running speed relatively fast at the same time. For example, the running times of DMSDL-QBSA-RF classification model for W1 and W2 are, respectively, 0.0504 seconds and 1.9292 seconds faster than the original RF classification model. Based on above results of data, the proposed classification model is better than the traditional RF and SVM model in oil layer classification. The comparison of oil layer classification result is shown in Figure 6, where, (a), (c), and (e) represent the actual oil layer distribution and (b), (d), and (f) represent DMSDL-QBSA-RF oil layer distribution. In addition, 0 means this depth has no oil or gas and 1 means this depth has oil or gas.

From Figure 6, we can see that the DMSDL-QBSA-RF classification model identifies that the oil layer distribution results are not much different from the test oil test results. It can accurately identify the distribution of oil and gas in a well. The DMSDL-QBSA-RF model is suitable for petroleum logging applications, which greatly reduces the difficulty of oil exploration and has a good application foreground.

Table 10: Comparison of numerical testing results on CEC2014 test sets ($F1$–$F15$, Dim = 10).

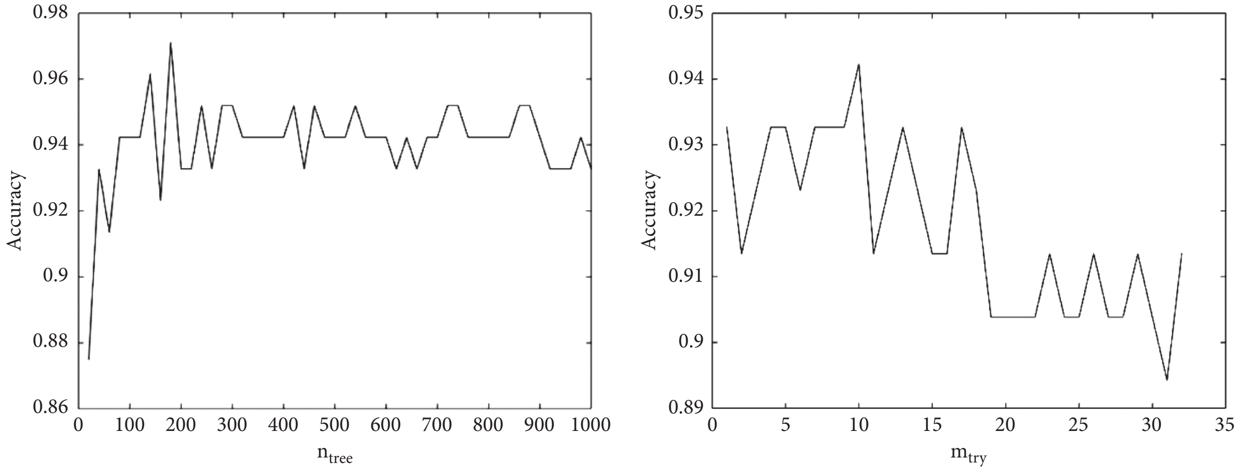
Function	Term	BSA	DE	DMSDL-PSO	DMSDL-BSA	DMSDL-QBSA
F_1	Max	**2.7411E+08**	3.6316E+09	6.8993E+08	3.9664E+08	9.6209E+08
	Min	1.4794E+07	3.1738E+09	1.3205E+06	**1.6929E+05**	1.7687E+05
	Mean	3.1107E+07	3.2020E+09	6.7081E+06	**1.3567E+06**	1.9320E+06
	Var	1.2900E+07	4.0848E+07	2.6376E+07	**1.0236E+07**	1.3093E+07
F_2	Max	**8.7763E+09**	1.3597E+10	1.3515E+10	1.6907E+10	1.7326E+10
	Min	1.7455E+09	1.1878E+10	3.6982E+04	2.1268E+05	**1.4074E+05**
	Mean	1.9206E+09	1.1951E+10	1.7449E+08	5.9354E+07	**4.8412E+07**
	Var	1.9900E+08	**1.5615E+08**	9.2365E+08	5.1642E+08	4.2984E+08
F_3	Max	4.8974E+05	**8.5863E+04**	2.6323E+06	2.4742E+06	5.3828E+06
	Min	1.1067E+04	1.4616E+04	1.8878E+03	1.2967E+03	**6.3986E+02**
	Mean	1.3286E+04	1.4976E+04	1.0451E+04	3.5184E+03	**3.0848E+03**
	Var	6.5283E+03	**1.9988E+03**	4.5736E+04	4.1580E+04	8.0572E+04
F_4	Max	**3.5252E+03**	9.7330E+03	4.5679E+03	4.9730E+03	4.3338E+03
	Min	5.0446E+02	8.4482E+03	**4.0222E+02**	4.0843E+02	4.1267E+02
	Mean	5.8061E+02	8.5355E+03	4.4199E+02	4.3908E+02	**4.3621E+02**
	Var	1.4590E+02	1.3305E+02	1.6766E+02	1.2212E+02	**8.5548E+01**
F_5	Max	5.2111E+02	5.2106E+02	**5.2075E+02**	5.2098E+02	5.2110E+02
	Min	**5.2001E+02**	5.2038E+02	5.2027E+02	5.2006E+02	5.2007E+02
	Mean	**5.2003E+02**	5.2041E+02	5.2033E+02	5.2014E+02	5.2014E+02
	Var	7.8380E−02	5.7620E−02	**5.6433E−02**	1.0577E−01	9.9700E−02
F_6	Max	**6.1243E+02**	6.1299E+02	6.1374E+02	6.1424E+02	6.1569E+02
	Min	6.0881E+02	6.1157E+02	6.0514E+02	6.0288E+02	**6.0257E+02**
	Mean	6.0904E+02	6.1164E+02	6.0604E+02	6.0401E+02	**6.0358E+02**
	Var	2.5608E−01	1.2632E−01	**1.0434E+00**	1.1717E+00	1.3117E+00
F_7	Max	**8.7895E+02**	1.0459E+03	9.1355E+02	1.0029E+03	9.3907E+02
	Min	7.4203E+02	1.0238E+03	**7.0013E+02**	7.0081E+02	7.0069E+02
	Mean	7.4322E+02	1.0253E+03	7.1184E+02	7.0332E+02	**7.0290E+02**
	Var	4.3160E+00	2.4258E+00	2.8075E+01	1.5135E+01	**1.3815E+01**
F_8	Max	**8.9972E+02**	9.1783E+02	9.6259E+02	9.2720E+02	9.3391E+02
	Min	8.4904E+02	8.8172E+02	8.3615E+02	8.1042E+02	**8.0877E+02**
	Mean	8.5087E+02	8.8406E+02	8.5213E+02	8.1888E+02	**8.1676E+02**
	Var	**2.1797E+00**	4.4494E+00	8.6249E+00	9.2595E+00	9.7968E+00
F_9	Max	1.0082E+03	**9.9538E+02**	1.0239E+03	1.0146E+03	1.0366E+03
	Min	9.3598E+02	9.6805E+02	9.2725E+02	9.2062E+02	**9.1537E+02**
	Mean	9.3902E+02	9.7034E+02	9.4290E+02	9.2754E+02	**9.2335E+02**
	Var	3.4172E+00	**3.2502E+00**	8.1321E+00	9.0492E+00	9.3860E+00
F_{10}	Max	**3.1087E+03**	3.5943E+03	3.9105E+03	3.9116E+03	3.4795E+03
	Min	2.2958E+03	2.8792E+03	1.5807E+03	1.4802E+03	**1.4316E+03**
	Mean	1.8668E+03	2.9172E+03	1.7627E+03	1.6336E+03	**1.6111E+03**
	Var	3.2703E+01	6.1787E+01	2.2359E+02	**1.9535E+02**	2.2195E+02
F_{11}	Max	3.6641E+03	**3.4628E+03**	4.0593E+03	3.5263E+03	3.7357E+03
	Min	2.4726E+03	2.8210E+03	1.7855E+03	1.4012E+03	**1.2811E+03**
	Mean	2.5553E+03	2.8614E+03	1.8532E+03	1.5790E+03	**1.4869E+03**
	Var	8.4488E+01	5.6351E+01	**1.3201E+02**	2.1085E+02	2.6682E+02
F_{12}	Max	1.2044E+03	1.2051E+03	1.2053E+03	1.2055E+03	**1.2044E+03**
	Min	1.2006E+03	1.2014E+03	1.2004E+03	**1.2003E+03**	1.2017E+03
	Mean	1.2007E+03	1.2017E+03	1.2007E+03	**1.2003E+03**	1.2018E+03
	Var	**1.4482E−01**	3.5583E−01	2.5873E−01	2.4643E−01	2.1603E−01
F_{13}	Max	**1.3049E+03**	1.3072E+03	1.3073E+03	1.3056E+03	1.3061E+03
	Min	1.3005E+03	1.3067E+03	1.3009E+03	**1.3003E+03**	1.3004E+03
	Mean	1.3006E+03	1.3068E+03	1.3011E+03	**1.3005E+03**	1.3006E+03
	Var	3.2018E−01	5.0767E−02	4.2173E−01	3.6570E−01	**2.9913E−01**
F_{14}	Max	**1.4395E+03**	1.4565E+03	1.4775E+03	1.4749E+03	1.4493E+03
	Min	1.4067E+03	1.4522E+03	1.4009E+03	**1.4003E+03**	1.4005E+03
	Mean	1.4079E+03	1.4529E+03	1.4024E+03	**1.4008E+03**	1.4009E+03
	Var	**2.1699E+00**	6.3013E−01	5.3198E+00	3.2578E+00	2.8527E+00
F_{15}	Max	5.4068E+04	**3.3586E+04**	4.8370E+05	4.0007E+05	1.9050E+05
	Min	1.9611E+03	2.1347E+04	1.5029E+03	1.5027E+03	**1.5026E+03**
	Mean	1.9933E+03	2.2417E+04	2.7920E+03	1.5860E+03	**1.5816E+03**
	Var	6.1622E+02	**1.2832E+03**	1.7802E+04	4.4091E+03	2.7233E+03

Table 11: Comparison of numerical testing results on CEC2014 test sets (F_{16}– F_{30}, Dim = 10).

Function	Term	BSA	DE	DMSDL-PSO	DMSDL-BSA	DMSDL-QBSA
F_{16}	Max	1.6044E+03	1.6044E+03	1.6046E+03	**1.6044E+03**	1.6046E+03
	Min	1.6034E+03	1.6041E+03	1.6028E+03	1.6021E+03	**1.6019E+03**
	Mean	1.6034E+03	1.6041E+03	1.6032E+03	1.6024E+03	**1.6024E+03**
	Var	**3.3300E−02**	3.5900E−02	2.5510E−01	3.7540E−01	3.6883E−01
F_{17}	Max	1.3711E+07	**1.6481E+06**	3.3071E+07	4.6525E+07	8.4770E+06
	Min	1.2526E+05	4.7216E+05	2.7261E+03	2.0177E+03	**2.0097E+03**
	Mean	1.6342E+05	4.8499E+05	8.7769E+04	3.1084E+04	**2.1095E+04**
	Var	2.0194E+05	**2.9949E+04**	1.2284E+06	8.7638E+05	2.0329E+05
F_{18}	Max	7.7173E+07	**3.5371E+07**	6.1684E+08	1.4216E+08	6.6050E+08
	Min	4.1934E+03	2.6103E+06	2.2743E+03	**1.8192E+03**	1.8288E+03
	Mean	**1.6509E+04**	3.4523E+06	1.2227E+06	1.0781E+05	1.9475E+05
	Var	7.9108E+05	**1.4888E+06**	2.1334E+07	3.6818E+06	1.0139E+07
F_{19}	Max	1.9851E+03	2.5657E+03	2.0875E+03	1.9872E+03	**1.9555E+03**
	Min	1.9292E+03	2.4816E+03	1.9027E+03	**1.9023E+03**	1.9028E+03
	Mean	1.9299E+03	2.4834E+03	1.9044E+03	**1.9032E+03**	1.9036E+03
	Var	**1.0820E+00**	3.8009E+00	1.1111E+01	3.3514E+00	1.4209E+00
F_{20}	Max	**8.3021E+06**	2.0160E+07	1.0350E+07	6.1162E+07	1.2708E+08
	Min	5.6288E+03	1.7838E+06	2.1570E+03	2.0408E+03	**2.0241E+03**
	Mean	1.2260E+04	1.8138E+06	5.6957E+03	**1.1337E+04**	4.5834E+04
	Var	**1.0918E+05**	5.9134E+05	1.3819E+05	6.4167E+05	2.1988E+06
F_{21}	Max	2.4495E+06	1.7278E+09	**2.0322E+06**	1.3473E+07	2.6897E+07
	Min	4.9016E+03	**1.4049E+09**	3.3699E+03	2.1842 E+03	2.2314E+03
	Mean	6.6613E+03	1.4153E+09	9.9472E+03	1.3972E+04	**7.4587E+03**
	Var	4.1702E+04	2.2557E+07	**3.3942E+04**	2.5098E+05	2.8735E+05
F_{22}	Max	**2.8304E+03**	4.9894E+03	3.1817E+03	3.1865E+03	3.2211E+03
	Min	2.5070E+03	4.1011E+03	2.3492E+03	2.2442E+03	**2.2314E+03**
	Mean	**2.5081E+03**	4.1175E+03	2.3694E+03	2.2962E+03	2.2687E+03
	Var	5.4064E+00	**3.8524E+01**	4.3029E+01	4.8006E+01	5.1234E+01
F_{23}	Max	2.8890E+03	**2.6834E+03**	2.8758E+03	3.0065E+03	2.9923E+03
	Min	2.5000E+03	2.6031E+03	**2.4870E+03**	2.5000E+03	2.5000E+03
	Mean	**2.5004E+03**	2.6088E+03	2.5326E+03	2.5010E+03	2.5015E+03
	Var	9.9486E+00	8.6432E+00	5.7045E+01	**1.4213E+01**	1.7156E+01
F_{24}	Max	**2.6293E+03**	2.6074E+03	2.6565E+03	2.6491E+03	2.6369E+03
	Min	2.5816E+03	2.6049E+03	2.5557E+03	**2.5246E+03**	2.5251E+03
	Mean	2.5829E+03	2.6052E+03	2.5671E+03	**2.5337E+03**	2.5338E+03
	Var	1.3640E+00	**3.3490E−01**	8.9434E+00	1.3050E+01	1.1715E+01
F_{25}	Max	2.7133E+03	**2.7014E+03**	2.7445E+03	2.7122E+03	2.7327E+03
	Min	2.6996E+03	2.7003E+03	2.6784E+03	2.6789E+03	**2.6635E+03**
	Mean	2.6996E+03	2.7004E+03	2.6831E+03	**2.6903E+03**	2.6894E+03
	Var	2.3283E−01	**9.9333E−02**	4.9609E+00	7.1175E+00	1.2571E+01
F_{26}	Max	**2.7056E+03**	2.8003E+03	2.7058E+03	2.7447E+03	2.7116E+03
	Min	2.7008E+03	2.8000E+03	2.7003E+03	2.7002E+03	**2.7002E+03**
	Mean	2.7010E+03	2.8000E+03	2.7005E+03	2.7003E+03	**2.7003E+03**
	Var	**3.1316E−01**	1.6500E−02	3.9700E−01	1.1168E+00	3.5003E−01
F_{27}	Max	**3.4052E+03**	5.7614E+03	3.3229E+03	3.4188E+03	3.4144E+03
	Min	2.9000E+03	3.9113E+03	2.9698E+03	2.8347E+03	**2.7054E+03**
	Mean	2.9038E+03	4.0351E+03	2.9816E+03	2.8668E+03	**2.7219E+03**
	Var	**3.2449E+01**	2.0673E+02	3.4400E+01	6.2696E+01	6.1325E+01
F_{28}	Max	4.4333E+03	5.4138E+03	4.5480E+03	**4.3490E+03**	4.8154E+03
	Min	3.0000E+03	4.0092E+03	3.4908E+03	3.0000E+03	**3.0000E+03**
	Mean	3.0079E+03	4.0606E+03	3.6004E+03	**3.0063E+03**	3.0065E+03
	Var	6.8101E+01	9.2507E+01	8.5705E+01	**4.2080E+01**	5.3483E+01
F_{29}	Max	**2.5038E+07**	1.6181E+08	5.9096E+07	7.0928E+07	6.4392E+07
	Min	3.2066E+03	1.0014E+08	3.1755E+03	**3.1005E+03**	3.3287E+03
	Mean	5.7057E+04	1.0476E+08	8.5591E+04	6.8388E+04	**2.8663E+04**
	Var	**4.9839E+05**	6.7995E+06	1.7272E+06	1.5808E+06	8.6740E+05

Table 11: Continued.

Function	Term	BSA	DE	DMSDL-PSO	DMSDL-BSA	DMSDL-QBSA
F_{30}	Max	$\mathbf{5.2623E+05}$	$2.8922E+07$	$1.1938E+06$	$1.2245E+06$	$1.1393E+06$
	Min	$5.9886E+03$	$1.9017E+07$	$3.7874E+03$	$\mathbf{3.6290E+03}$	$3.6416E+03$
	Mean	$7.4148E+03$	$2.0002E+07$	$5.5468E+03$	$4.3605E+03$	$\mathbf{4.1746E+03}$
	Var	$\mathbf{1.1434E+04}$	$1.1968E+06$	$3.2255E+04$	$2.2987E+04$	$1.8202E+04$



(a)

(b)

Figure 3: The effect of the two parameters on the performance of RF models: (a) the effect of n_{tree}; (b) the effect of m_{try}.

```
Variable setting: number of iterations: iter, bird positions: $X_i$, local optimum: $P_i$, global optimal position: gbest, global optimum:
    $P_{\text{gbest}}$, and out of bag error: OOB error;
Input: training dataset: Train, Train_Label, test dataset: Test, the parameters of DMSDL-QBSA;
Output: the label of test dataset: Test_Label, the final classification model: DMSDL-QBSA-RF;
Begin
/*Build classification model based on DMSDL-QBSA-RF*/
Initialize the positions of N birds using equations (16) and (17): $X_i\,(i = 1, 2, \ldots, N)$;
    Calculated fitness: $f(X_i)\,(i = 1, 2, \ldots, N)$; set $X_i$ to be $P_i$ and find $P_{\text{gbest}}$;
While iter < $\text{iter}_{\max}$ + 1 do
        For $i = 1$: $n_{\text{tree}}$
            Give each tree a training set of size $N$ by random sampling with replacement based on Bootstrap;
        Select $m_{\text{try}}$ attributes randomly at each leaf node, compare the attributes, and select the best one;
            Recursively generate each decision tree without pruning operations;
        End For
        Update classification accuracy of RF: Evaluate $f(X_i)$;
        Update gbest and $P_{\text{gbest}}$;
            $[n_{\text{best}}, m_{\text{best}}]$ = gbest;
            iter = iter + 1;
    End While
/*Classify using RF model*/
        For $i = 1$: $n_{\text{best}}$
            Give each tree a training set of size $N$ by random sampling with replacement based on Bootstrap;
        Select $m_{\text{best}}$ attributes randomly at each leaf node, compare the attributes, and select the best one;
            Recursively generate each decision tree without pruning operations;
        End
        Return DMSDL-QBSA-RF classification model
        Classify the test dataset using equation (20);
        Calculate OOB error;
End
```

Algorithm 2: DMSDL-QBSA-RF classification model.

Table 12: Comparison of numerical testing results on eight UCI datasets.

Dataset	RF (%)	BSA-RF (%)	DMSDL-BSA-RF (%)	DMSDL-QBSA-RF (%)
Blood	75.40	80.80	79.46	**81.25**
Heart-statlog	82.10	**86.42**	**86.42**	**86.42**
Sonar	78.39	85.48	85.48	**88.71**
Appendicitis	83.23	87.10	90.32	**93.55**
Cleve	79.55	87.50	88.64	**89.77**
Magic	88.95	89.85	**90.25**	89.32
Mammographic	74.73	77.68	76.79	**79.46**
Australian	85.83	88.84	88.84	**90.78**

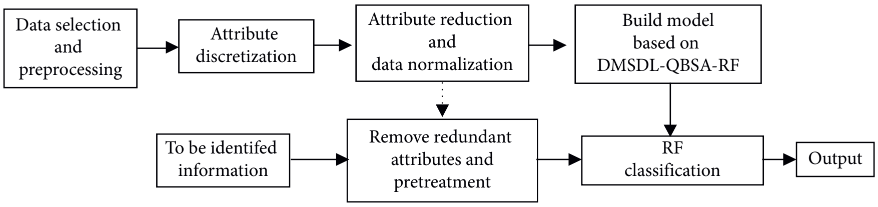

Data selection and preprocessing
Attribute discretization
Attribute reduction and data normalization
Build model based on DMSDL-QBSA-RF
To be identifed information
Remove redundant attributes and pretreatment
RF classification
Output

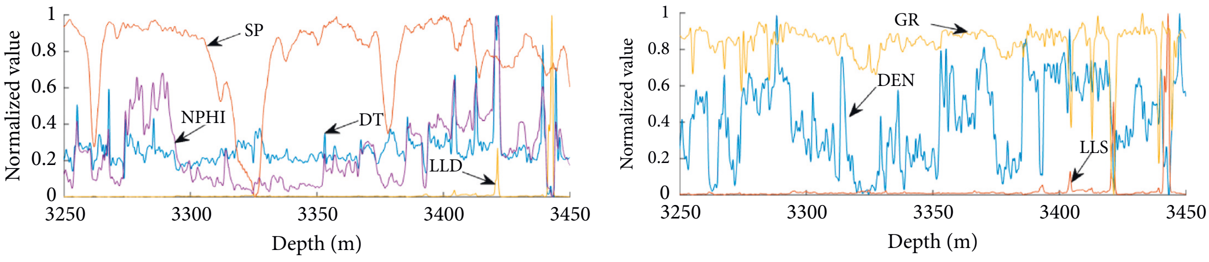
Figure 4: Block diagram of the oil layer classification system based on DMSDL-QBSA-RF.

Table 13: Distribution of the logging data.

Well	Training dataset				Test dataset			
	Depth (m)	Data	Oil/gas layers	Dry layer	Depth (m)	Data	Oil/gas layers	Dry layer
*W*1	3027~3058	250	203	47	3250~3450	1600	237	1363
*W*2	2642~2648	30	10	20	2940~2978	191	99	92
*W*3	1040~1059	160	47	113	1120~1290	1114	96	1018

Table 14: Attribute reduction results of the logging dataset.

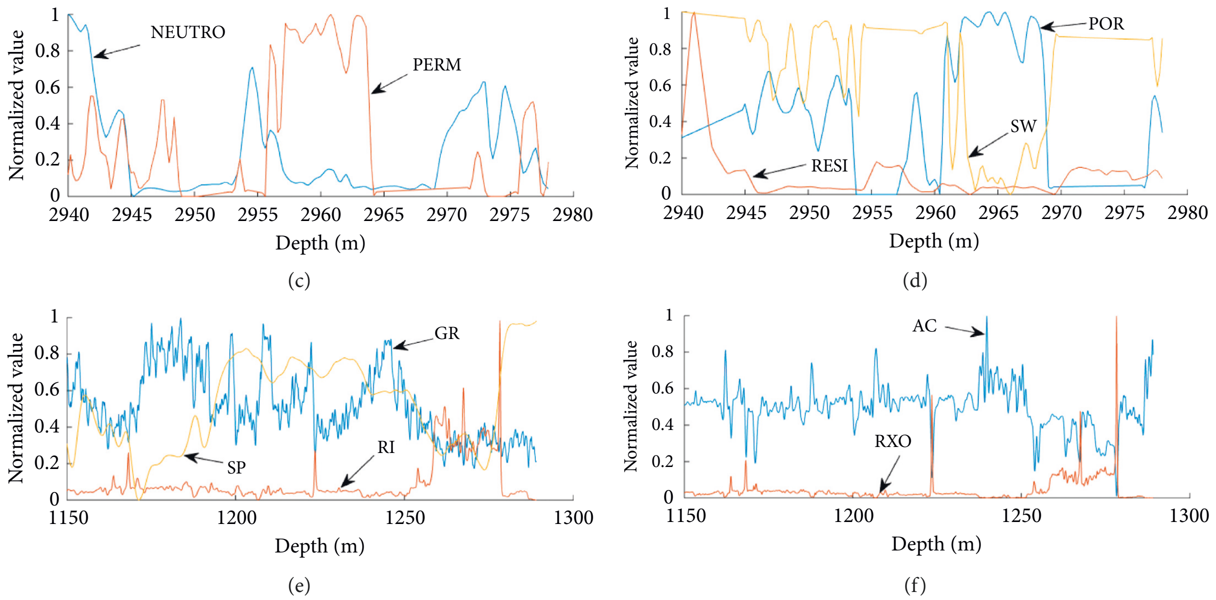
Well		Attributes
*W*1	Actual attributes	{GR, DT, SP, WQ, LLD, LLS, DEN, NPHI, PE, U, TH, K, CALI}
	Reduction attributes	{GR, DT, SP, LLD, LLS, DEN, NPHI}
*W*2	Actual attributes	{DENSITY, GAMM, VCLOK, NEUTRO, PERM, POR, RESI, SONIC, SP, SW}
	Reduction attributes	{NEUTRO, PERM, POR, RESI, SW}
*W*3	Actual attributes	{AC, CNL, DEN, GR, RT, RI, RXO, SP, R2M, R025, BZSP, RA2, C1, C2, CALI, RINC, PORT, VCL, VMA1, VMA6, RHOG, SW, VO, WO, PORE, VXO, VW, AC1}
	Reduction attributes	{AC, GR, RI, RXO, SP}



(a)

(b)

Figure 5: Continued.



(c)

(d)

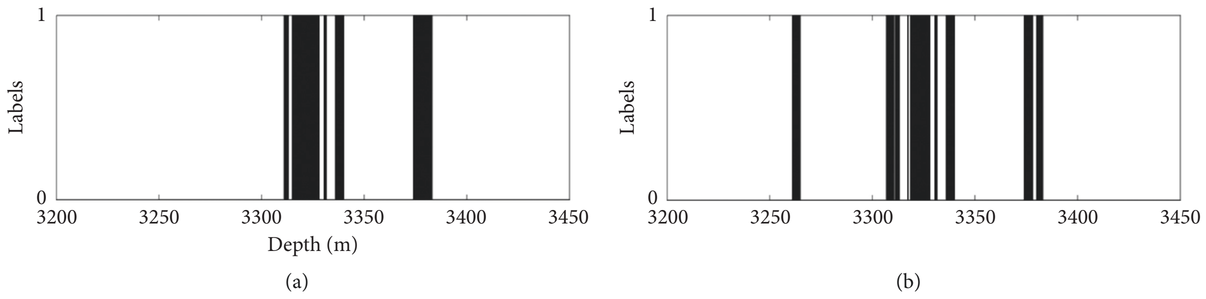
(e)

(f)

Figure 5: The normalized curves of attributes: (a) and (b) attribute normalization of *W*1; (c) and (d) attribute normalization of *W*2; (e) and (f) attribute normalization of *W*3.

Table 15: Performance of various well data.

Well	Classification model	RMSE	MAE	Accuracy (%)	Running time (s)
W1	RF	0.3326	0.1106	88.94	1.5167
	SVM	0.2681	0.0719	92.81	4.5861
	BSA-RF	0.3269	0.1069	89.31	1.8064
	DMSDL-BSA-RF	0.2806	0.0788	92.13	2.3728
	DMSDL-QBSA-RF	**0.2449**	**0.0600**	**94.00**	**1.4663**
W2	RF	0.4219	0.1780	82.20	3.1579
	SVM	0.2983	0.0890	91.10	4.2604
	BSA-RF	0.3963	0.1571	84.29	1.2979
	DMSDL-BSA-RF	0.2506	0.062827	93.72	1.6124
	DMSDL-QBSA-RF	**0.2399**	**0.0576**	**94.24**	**1.2287**
W3	RF	0.4028	0.1622	83.78	2.4971
	SVM	0.2507	0.0628	93.72	2.1027
	BSA-RF	0.3631	0.1318	86.81	1.3791
	DMSDL-BSA-RF	0.2341	0.0548	94.52	**0.3125**
	DMSDL-QBSA-RF	**0.0519**	**0.0027**	**99.73**	0.9513



(a)

(b)

Figure 6: Continued.

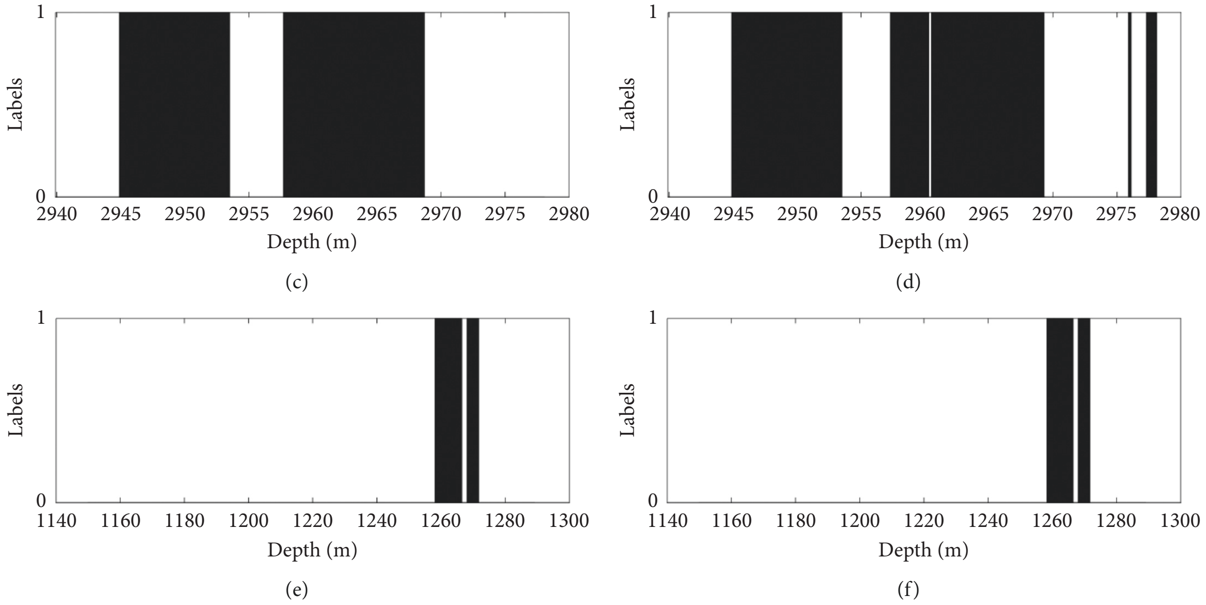


Figure 6: Classification of DMSDL-QBSA-RF: (a) the actual oil layer distribution of W1; (b) DMSDL-QBSA-RF oil layer distribution of W1; (c) the actual oil layer distribution of W2; (d) DMSDL-QBSA-RF oil layer distribution of 2; (e) the actual oil layer distribution of 3; (f) DMSDL-QBSA-RF oil layer distribution of 3.

5. Conclusion

This paper presents an improved BSA called DMSDL-QBSA, which employed the dynamic multi-swarm method, differential evolution, and quantum behavior to enhance the global and the local exploration capabilities of original BSA. First, 18 classical benchmark functions are used to verify the effectiveness of the improved method. The experimental study of the effects of these three strategies on the performance of DMSDL-QBSA revealed that the hybrid method has an excellent influence to improve the improvement of original GOA and especially original DE. Second, compared with the popular intelligence algorithms, such as GWO, WOA, SCA, GOA, and SSA, the DMSDL-QBSA can provide more competitive results on the 18 classical benchmark functions. Additionally, 30 complex CEC2014 test functions are used to better verify the performance of DMSDL-QBSA in a more comprehensively manner. The DMSDL-QBSA can show more excellent performance on the 18 classical benchmark functions. Finally, the improved DMSDL-QBSA is used to optimize the parameters of RF. Experimental results on actual oil logging prediction problem have proved that the classification accuracy of the established DMSDL-QBSA-RF classification model can get 94.00%, 94.24%, and 99.73% on these wells, and the accuracy is much higher than the original RF model. At the same time, the running speed performed faster than other four advanced classification models on most wells.

Although the proposed DMSDL-QBSA has been proven to be effective in solving general optimization problems, DMSDL-QBSA has some shortcomings that warrant further investigation. And in DMSDL-QBSA, due to the hybrid of three strategies, DMSDL-QBSA has needed more time than the classical BSA. Therefore, deploying the proposed algorithm to increase recognition efficiency is a worthwhile direction. In the future research work, the method presented in this paper can also be extended to solving discrete optimization problems and multiobjective optimization problems. Furthermore, applying the proposed DMSDL-QBSA-RF model to other fields such as financial prediction and biomedical science diagnosis is also an interesting future work.

Data Availability

All data included in this study are available upon request by contact with the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. U1813222), Tianjin Natural Science Foundation, China (no. 18JCYBJC16500), Key Research and Development Project from Hebei Province, China (nos. 19210404D and 20351802D), Key Research Project of Science and Technology from the Ministry of Education of Hebei Province, China (no. ZD2019010), and Project of Industry-University Cooperative Education of Ministry of Education of China (no. 201801335014).

References

[1] A. P. Engelbrecht, *Foundations of Computational Swarm Intelligence*, Beijing: Tsinghua University Press, Beijing, China, 2019.

[2] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of the IEEE International Conference Neural Networks*, Perth, Australia, July 1995.

[3] D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” Technical report-TR06, Erciyes University, Kayseri, Turkey, 2005.

[4] X. Li, *A new intelligent optimization method-artificial fish school algorithm*, Doctoral thesis, Zhejiang University, Hangzhou, China, 2003.

[5] X. S. Yang, “Firefly algorithms for multimodal optimization stochastic algorithms: foundations and applications,” in *Proceedings of the 5th International Symposium on Stochastic Algorithms*, SAGA, Berlin, Germany, 2009.

[6] Y. Sharafi, M. A. Khanesar, and M. Teshnehlab, “Discrete binary cat swarm optimization,” in *Proceedings of the IEEE International Conference on Computer, Control and Communication (IC4)*, Karachi, Pakistan, September 2013.

[7] X. B. Meng, X. Z. Gao, L. Lu, Y. Liu, and H. Z. Zhang, “A new bio-inspired optimisation algorithm: bird swarm algorithm,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 28, no. 4, pp. 673–687, 2016.

[8] N. Jatana and B. Suri, “Particle swarm and genetic algorithm applied to mutation testing for test data generation: a comparative evaluation,” *Journal of King Saud University-Computer and Information Sciences*, vol. 32, pp. 514–521, 2020.

[9] H. Chung and K. Shin, “Genetic algorithm-optimized multichannel convolutional neural network for stock market prediction,” in *Neural Computing and Applications*, Springer, New York, NY, USA, 2019.

[10] I. Strumberger, E. Tuba, M. Zivkovic, N. Bacanin, M. Beko, and M. Tuba, “Designing convolutional neural network architecture by the firefly algorithm,” in *Proceedings of the 2019 IEEE International Young Engineers Forum (YEF-ECE)*, Costa da Caparica, Portugal, May, 2019.

[11] I. Strumberger, N. Bacanin, M. Tuba, and E. Tuba, “Resource scheduling in cloud computing based on a hybridized whale optimization algorithm,” *Applied Sciences*, vol. 9, no. 22, p. 4893, 2019.

[12] M. Tuba and N. Bacanin, “Improved seeker optimization algorithm hybridized with firefly algorithm for constrained optimization problems,” *Neurocomputing*, vol. 143, pp. 197–207, 2014.

[13] I. Strumberger, M. Minovic, M. Tuba, and N. Bacanin, “Performance of elephant herding optimization and tree growth algorithm adapted for node localization in wireless sensor networks,” *Sensors*, vol. 19, no. 11, p. 2515, 2019.

[14] X. S. Yang, “Swarm intelligence based algorithms: a critical analysis,” *Evolutionary Intelligence*, vol. 7, no. 1, pp. 17–28, 2014.

[15] N. Bacanin and M. Tuba, “Artificial bee colony (ABC) algorithm for constrained optimization improved with genetic operators,” *Studies in Informatics and Control*, vol. 21, no. 2, pp. 137–146, 2012.

[16] J. Liu, H. Peng, Z. Wu, J. Chen, and C. Deng, *Multi-Strategy Brainstorm Optimization Algorithm with Dynamic Parameters Adjustment*, Applied Intelligence, Guildford, UK, 2010.

[17] H. Peng, Y. He, and C. Deng, “Firefly algorithm with luciferase inhibition mechanism,” *IEEE Access*, vol. 7, pp. 120189–120201, 2019.

[18] H. Peng, C. Deng, and Z. Wu, “Best neighbor guided artificial bee colony algorithm for continuous optimization problems,” *Soft Computing*, vol. 23, no. 18, pp. 8723–8740, 2019.

[19] C. Xu and R. Yang, “Parameter estimation for chaotic systems using improved bird awarm algorithm,” *Modern Physics Letters B*, vol. 31, no. 36, pp. 1–15, 2017.

[20] J. Yang and Y. Liu, “Separation of signals with same frequency based on improved bird swarm algorithm,” in *Proceedings of the International Symposium on Distributed Computing and Applications for Business Engineering and Science*, Wuxi, China, August 2018.

[21] X. Wang, Y. Deng, and H. Duan, “Edge-based target detection for unmanned aerial vehicles using competitive bird swarm algorithm,” *Aerospace Science & Technology*, vol. 78, pp. 708–720, 2018.

[22] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[23] L. Ma and S. Fan, “CURE-SMOTE algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests,” *BMC Bioinformatics*, vol. 18, no. 1, pp. 1–18, 2017.

[24] J. Hou, Q. Li, and S. Meng, “A differential privacy protection random forest,” *IEEE Access*, vol. 7, pp. 130707–130720, 2019.

[25] G. L. Liu, S. J. Han, and X. H. Zhao, “Optimisation algorithms for spatially constrained forest planning,” *Ecological Modelling*, vol. 194, no. 4, pp. 421–428, 2006.

[26] D. Gong, B. Xu, and Y. Zhang, “A similarity-based cooperative co-evolutionary algorithm for dynamic interval multiobjective optimization problems,” in *Proceedings of the IEEE Transactions on Evolutionary Computation*, Piscataway, NJ, USA, February 2019.

[27] M. Rong, D. Gong, and W. Pedrycz, “A multi-model prediction method for dynamic multi-objective evolutionary optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 24, pp. 290–304, 2020.

[28] B. Xu, Y. Zhang, D. Gong, Y. Guo, and M. Rong, “Environment sensitivity-based cooperative co-evolutionary algorithms for dynamic multi-objective optimization,” *IEEE-ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 6, pp. 1877–1890, 2018.

[29] Y. Guo, H. Yang, M. Chen, J. Cheng, and D. Gong, “Ensemble prediction-based dynamic robust multi-objective optimization methods,” *Swarm and Evolutionary Computation*, vol. 48, pp. 156–171, 2019.

[30] J. J. Liang and P. N. Suganthan, “Dynamic multi-swarm particle swarm optimizer with local search,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, Edinburgh, Scotland, 2005.

[31] Y. Chen, L. Li, and H. Peng, “Dynamic multi-swarm differential learning particle swarm optimizer,” *Swarm and Evolutionary Computation*, vol. 39, pp. 209–221, 2018.

[32] R. Storn, “Differential evolution design of an IIR-filter,” in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC 96)*, Nagoya, Japan, May 1996.

[33] Z. Liu, X. Ji, and Y. Yang, “Hierarchical differential evolution algorithm combined with multi-cross operation,” *Expert Systems with Applications*, vol. 130, pp. 276–292, 2019.

[34] H. Peng, Z. Guo, and C. Deng, “Enhancing differential evolution with random neighbors based strategy,” *Journal of Computational Science*, vol. 26, pp. 501–511, 2018.

[35] J. Sun, W. Fang, X. Wu, V. Palade, and W. Xu, “Quantum-behaved particle swarm optimization: analysis of individual particle behavior and parameter selection,” *Evolutionary Computation*, vol. 20, no. 3, pp. 349–393, 2012.

[36] B. Chen, H. Lei, H. Shen, Y. Liu, and Y. Lu, "A hybrid quantum-based PIO algorithm for global numerical optimization," *Science China–Information Sciences*, vol. 62, no. 7, 2019.

[37] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.

[38] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[39] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[40] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge Based Systems*, vol. 95, pp. 120–133, 2016.

[41] S. Shahrzad, M. Seyedali, and L. Andrew, "Grasshopper optimisation algorithm: theory and application," *Advances in Engineering Software*, vol. 105, pp. 30–47, 2017.

[42] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," *Systems Science & Control Engineering An Open Access Journal*, vol. 8, no. 1, pp. 22–34, 2020.

[43] J. Bai, K. Xia, Y. Lin, and P. Wu, "Attribute reduction based on consistent covering rough set and its application," *Complexity*, vol. 2017, Article ID 8986917, 9 pages, 2017.