IBDA: Improved Binary Dragonfly Algorithm With Evolutionary Population Dynamics and Adaptive Crossover for Feature Selection

JIAHUI LI[ID][1,2], **HUI KANG**[ID][1,2], **GENG SUN**[ID][1,2], **(Member, IEEE), TIE FENG**[ID][1,2], **WENQI LI**[ID][1,2], **WEI ZHANG**[3], **AND BAI JI**[3]

[1]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun 130012, China
[3]Department of Hepatobiliary and Pancreatic Surgery, First Hospital, Jilin University, Changchun 130021, China

Corresponding author: Geng Sun (sungeng@jlu.edu.cn)

ABSTRACT Feature selection is an effective method to eliminate irrelevant, redundant and noisy features, which improves the performance of classification and reduces the computational burden in machine learning. In this paper, an improved binary dragonfly algorithm (IBDA) which extends from the conventional dragonfly algorithm (DA) is proposed as a search strategy to design a wrapper-based feature selection method. First, a novel evolutionary population dynamics (EPD) strategy is introduced in IBDA to enhance the exploitation ability while ensuring population diversity of the algorithm. Second, IBDA proposes a novel crossover operator which establishes connections between the crossover rates and iterations so that making the algorithm can adjust the crossover rates of solutions dynamically, thereby balancing the exploitation and exploration of the algorithm. Finally, a binary mechanism is proposed to make the algorithm suitable for the binary feature selection problems. Simulations are conducted on 27 classical datasets from the UC Irvine Machine Learning Repository, and the results demonstrate that the proposed IBDA has better performance than some other comparison algorithms. Moreover, the effectiveness and performance of the proposed improved factors are evaluated by tests.

INDEX TERMS Feature selection, classification, bio-inspired computing, dragonfly algorithm.

I. INTRODUCTION

The availability of large scale datasets has boosted the applications of machine learning in many fields, such as active matter [1], molecular and materials science [2], and biomedical [3]. With the increasing of complexities of the machine learning models, more and more datasets with high-dimensional feature spaces are generated. However, part of the features are irrelevant and redundant, which may reduce the classification accuracy and waste the computing resources.

Feature selection is an effective method to overcome the abovementioned issues [4]. The main idea of feature selection is to select the informative subset from a high-dimensional feature space [5], such that the number of features can be reduced and the worthless features can be deleted, thereby saving the computing resources and increasing the classification accuracy for machine learning.

Feature selection methods are mainly divided into two categories that are filter and wrapper approaches. The filter-based method assigns a relevance score to each feature by using a statistical measure. Then, it ranks the features according to the calculated scores and selects the subset of features depending on a user-defined criterion [6]. For the wrapper-based approach, it utilizes a classifier to guide the feature selection results and the accuracies of this method are usually better than the filter-based method. Thus, the wrapper-based approach is an efficient method for feature

The associate editor coordinating the review of this manuscript and approving it for publication was Donato Impedovo[ID].

selection. However, this method may take more computing resources and it closely related to the learning algorithms [7].

Feature selection can be considered as a global combination optimization problem in nature. Thus, it can be solved by swarm intelligence algorithms especially for the wrapper-based method. Some researchers have adopted several swarm intelligence algorithms such as genetic algorithm (GA) [8], particle swarm optimization (PSO) [9], ant colony optimization (ACO) [10], bat algorithm (BA) [11], gray wolf optimization (GWO) [12], and the variants of these algorithms for the feature selection problems.

DA is a novel swarm intelligence algorithm proposed by Mirjalili [13] in 2016 for solving continuous optimization problems, and it performs better performance compared to some other approaches [14] due to the effectiveness and accessibility. However, according to the no free lunch (NFL) theory, no algorithm can solve all optimization problems suitably. Moreover, the conventional DA can not be directly used for feature selection problems since these problems are with binary solution spaces. In addition, DA may have some certain shortcomings, e.g., its exploitation ability relies on the sub-swarm mechanism, which is insufficient for exploiting high-dimensional solution spaces of feature selection. Thus, it is necessary to improve conventional DA to make it more suitable for feature selection.

The main contributions of this paper are summarized as follows:

1) We formulate an optimization problem to jointly reduce the number of selected features and improve the classification accuracy.
2) We propose an improved binary DA (IBDA) to solve the formulated joint feature selection problem. IBDA introduces a novel evolutionary population dynamics (EPD) mechanism, an adaptive crossover (AC) factor and a binary strategy to improve the performance of conventional DA and make it more suitable for feature selection.
3) Experiments based on the UC Irvine Machine Learning Repository are conducted to evaluate the performance of the proposed IBDA for feature selection, and the results are compared to some other algorithms. Moreover, the effectiveness of the proposed improved factors are verified.

The rest of this paper is organized as follows. Section II reviews the related work. Section III formulates the joint feature selection problem. Section IV proposes the IBDA. Section V shows the experiment results and Section VI presents a summary of findings and conclusions.

II. RELATED WORK
Heuristic algorithms can solve various optimization problems including the feature selection. Due to their effectiveness and simplicity, many heuristic algorithms have been proposed for solving the feature selection problems, e.g., GA [15], PSO [16], GWO [17], flower pollination

algorithm (FPA) [18], artificial bee colony (ABC) [19], bacterial foraging optimization (BFO) [20], BA [21], cuckoo search (CS) [22], firefly algorithm (FA) [23], whale optimization algorithm (WOA) [24], grasshopper optimization algorithm (GOA) [25]. Recently, more and more heuristic algorithms are proposed to deal with many kinds of optimization problems. For instance, inspired by the navigation and foraging behaviors of sales, Mirjalili *et al.* [26] propose a salp swarm algorithm (SSA) for the airfoil and marine propeller design problems. Moreover, Dhiman and Kumar [27] propose a novel nature-inspired algorithm called emperor penguin optimizer (EPO), which mimics the huddling behavior of emperor penguins. Seagull optimization algorithm (SOA) is another heuristic algorithm, which is inspired by the migration and attacking behaviors of seagulls [28]. In addition, Anita and Yadav [29] propose a novel artificial electric field algorithm (AEFA) which inspired by the Coulomb's law of electrostatic force. More heuristic algorithms can be found in literatures [30], [31] and [32].

Many heuristic optimization algorithms have been used as the search strategies in wrapper-based feature selection methods, and some representative approaches are summarized and reviewed as follows.

García-Dominguez *et al.* [33] utilize GA for feature selection to reduce the original size of the environmental sound data. Liu and Shang [34] propose a fast wrapper feature subset selection algorithm based on PSO, which employs the domain knowledge of feature subset selection problems. Reference [35] uses a binary BA for feature selection to reduce the size of stego and cover images data. Devanathan *et al.* [36] exploit a binary GWO (BGWO)-based feature selection method and the bag-of-keypoint features (BoKF) model to distinguish nucleolar and centromere staining patterns. Rodrigues *et al.* [37] use the CS algorithm to solve the feature selection problems in two datasets obtained from a Brazilian electrical power company. In [38], a feature selection algorithm based on the moth-flame optimization (MFO) is proposed. Moreover, Emary *et al.* [12] propose a novel binary version of GWO and use it to select optimal feature subset for classification purposes.

Recently, some improved approaches that combining different swarm intelligence algorithms or introducing enhanced factors are proposed for feature selection problems. For example, the authors in [39] propose a new algorithm by combining the differential evolution (DE) and ABC algorithms for feature selection. Reference [40] propose a hybrid algorithm called ACO-ABC to solve the feature selection problems. Al-Tashi *et al.* [41] propose a binary version of the hybrid GWO-PSO algorithm for selecting features. Bharti and Bharti [42] select an informative subset of features by employing a crossbreed approach of binary PSO (BPSO) and sine cosine algorithm (SCA). The authors in reference [43] propose an enhanced hybrid metaheuristic approach by combining GWO and WOA to develop a wrapper-based feature selection method. For the algorithms with improved factors, Zhang *et al.* [44] proposed

a return-cost-based binary firefly algorithm (Rc-BBFA) for the feature selection problems. The authors in reference [45] use eight transfer functions and the crossover operator to enhance the exploratory behavior of SSA for feature selections. Reference [46] propose an improved version of gravitational search algorithm (GSA) by using the concept of global memory and the definition of exponential K_{best} to solve the feature selection problems. Tumar *et al.* [47] propose an enhanced binary moth flame optimization (EBMFO) with adaptive synthetic sampling (ADASYN) to predict the most optimal feature combination in software faults. More improved algorithms for feature selections can be found in literature [48]–[53], and [54].

There are also several previous works that focus on DA-based methods and their applications in feature selections. Mafarja *et al.* [55] propose a wrapper-feature selection algorithm based on the binary DA (BDA). For example, Elhariri *et al.* [56] solve the problem of electromyography (EMG) signal classification with optimal features subset selection by using DA and support vector machines (SVM) classification. Tawhid and Dsouza [57] combine the BDA and enhanced PSO to propose a hybrid BDA-enhanced PSO (HBDESPO) algorithm for feature selections. In reference [58], a combination of wavelet packet-based features and improved binary dragonfly optimization-based feature selection method is proposed to classify different types of infant cry signals. Moreover, Sayed *et al.* [59] propose a chaotic DA (CDA) where the chaotic maps are embedded with the searching iterations of the algorithm for feature selections.

The above methods can solve the problem of feature selection in various applications. However, an optimization algorithm may perform different performance in different applications. Thus, the existing methods cannot solve all feature selection problems properly, and this motives us to propose an IBDA with suitable improved factors to deal with more feature selection problems in this work.

III. PROBLEM FORMULATION

In this work, we aim to select part features from the whole dataset so that achieving the following two objectives: (1) reducing the number of selected features, (2) enhancing the classification accuracy. Thus, to simultaneously achieve these two goals, we design a joint fitness function as follows:

$$f_{Fitness} = \alpha \cdot E_r + \beta \cdot \frac{F_s}{F_a} \tag{1}$$

where F_s and F_a are the number of selected features and total number of features, respectively. Moreover, $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$ are the weights of these two objectives, respectively. E_r indicates the classification error rate of a certain classifier.

As can be seen, the fitness function consists of two parts, the first part is used to guarantee the classification accuracy while the second part is used to reduce the number of selected features, and they are combined by using the linear

weighting method. These two objectives can be adjusted according to different feature selection problem to obtain the results with different biases. Specifically, if α is turned up while β is turned down, then the approach that utilizes this fitness function tends to select more features to obtain higher accuracy. Conversely, if α is turned with smaller value and β is with larger value, the algorithm tends to sacrifice partial accuracy to reduce the number of selected features.

Note that using a simple and relatively cheap classification algorithm in a wrapper approach, such as k-nearest neighbor (KNN) and decision tree (DT), can obtain a good feature subset that is also suitable for the complex classification algorithms [60]. Thus, the KNN [61] method is introduced as a classifier in this paper since it is effective and easy to be implemented.

IV. ALGORITHM

The solution space of the formulated feature selection problem is discrete and relatively huge especially when the datasets are with large numbers of features, which is difficult to be solved. Thus, we propose an IBDA with the EPD mechanism, crossover strategy and binary scheme to make it more suitable for the feature selection problems.

A. CONVENTIONAL DA

DA is inspired by dynamic and static behaviors of swarming of dragonflies in nature. In the static swarm, dragonflies create sub-swarms and fly over small areas to hunt foods. In dynamic swarms, dragonflies make the swarm for migrating in one direction over long distances. Due to the nature of them, static and dynamic behaviors can represent the exploitation and exploration optimization phases, respectively.

There are three primitive principles of swarming behavior to simulate the behaviors of dragonflies, that are: Separation (S), alignment (A) and cohesion (C) are the individual's behaviors affected by the sub-swarm (a certain area around each dragonfly). Moreover, the main objective of DA is to make any swarm to be survival, such that all of the individuals should be attracted towards food sources (F) and distracted outward enemies (E).

The main factors that related to the solution update method in DA are introduced as follows.

1) The separation is expressed as:

$$S_i = - \sum_{j=1}^{N_{sub}} (X - X_m) \tag{2}$$

where S_i indicates the separation of the ith individual, X is defined as the individual's current position, X_m shows the position of the mth individual of the sub-swarm and N_{sub} denotes the number of individuals of the sub-swarm.

2) The alignment is calculated as follows:

$$A_i = \frac{\sum_{j=1}^{N_{sub}} V_m}{N_{sub}} \quad (3)$$

where V_m shows the velocity of the mth individual of the sub-swarm.

3) The cohesion is calculated as follows:

$$C_i = \frac{\sum_{j=1}^{N_{sub}} X_m}{N_{sub}} - X \quad (4)$$

4) The attraction towards a food source in DA is expressed as follows:

$$Food_i = X^+ - X \quad (5)$$

where X^+ shows the position of the food source.

5) The distraction outwards an enemy is calculated as follow:

$$Emery_i = X^- + X \quad (6)$$

where X^- shows the position of the enemy.

Accordingly, the step vector that combines the abovementioned five behaviors for updating the positions of dragonflies is defined as follows:

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fFood_i + eEmery_i) + w\Delta X_t \quad (7)$$

where s, a, c, f, e, and w indicate the separation weight, alignment weight, cohesion weight, food factor, enemy factor, and inertia weight, respectively. Moreover, t is the iteration counter.

Then, the individual position vector based on the step vector is defined as follows:

$$X_{t+1} = X_t + \Delta X_{t+1} \quad (8)$$

Moreover, DA uses Lévy flight mechanism as the random walk factor to enhance the stochastic behaviour and exploration ability of dragonflies. The position update method of dragonflies by using Lévy flight is expressed as follows:

$$X_{t+1} = X_t + Lévy \otimes X_t \quad (9)$$

The Lévy flight is calculated as follows:

$$Lévy = \frac{\mu}{|v|^{\frac{1}{\eta}}} \quad (10)$$

where $\mu \sim N(0, \sigma^2)$, $v \sim N(0, 1)$, η is a constant (equal to 1.5 in this work), and σ is calculated as follows:

$$\sigma = \left\{ \frac{\Gamma(1+\eta) \times sin(\frac{\pi\eta}{2})}{\eta \times \Gamma(\frac{1+\eta}{2}) \times 2^{\frac{\eta-1}{2}}} \right\}^{\frac{1}{\eta}} \quad (11)$$

where $\Gamma(y) = (y-1)!$.

Accordingly, the pseudo-code of the conventional DA is shown in Algorithm 1.

Algorithm 1 Conventional DA

1 Define and initialize the related parameters: swarm size N_{swarm}, solution dimension N_{dim}, maximum iteration N_{max_iter}, and fitness function, etc.;
2 **for** $t = 1$ *to* N_{max_iter} **do**
3   Calculate the fitness function values of dragonflies;
4   Update the food source and enemy, namely, *food_position* and *enemy_position*;
5   Update w, s, a, c, f, and e;
6   **for** $i = 1$ *to* N_{swarm} **do**
7   **if** *No other dragonflies in the sub-swarm* **then**
8   Perform random walk by using Eqs. (9);
9   **end**
10   **else**
11   Calculate $S_i, A_i, C_i, Food_i, Emery_i$ by using Eqs. (2), (3), (4), (5), (6);
12   Calculate the step vector by using Eq. (7);
13   Update the position vector of ith dragonfly by using Eq. (8);
14   **end**
15   **end**
16 **end**
17 Return *food_position*;
18 //*food_position* **is the best solution obtained by the algorithm**

B. IBDA

It has been demonstrated that conventional DA has certain shortcomings for some optimization problems. For example, it may lack exploitation capabilities for the optimization problems with huge solution space. Moreover, conventional DA is originally proposed for the continuous optimization problems. However, feature selection is a discrete optimization problem since the solution space is binary. Thus, an IBDA is proposed for solving the formulated feature selection problem. IBDA introduces a novel EPD mechanism, a crossover strategy and a binary mechanism to improve the performance of the conventional DA and make it suitable for feature selection.

The main steps of IBDA is shown in Algorithm 2. Note that since the distance of dragonflies cannot be determined clearly in a binary space, the IBDA considers all of the dragonflies as one sub-swarm. Moreover, the details of the proposed factors are introduced as follows.

1) EPD

EPD is an evolutionary operator which is based on the theory of self-organized criticality (SOC) [62]. The purpose of EPD is to eliminate the worst individuals in the swarm by repositioning them around the best solutions, so that improving exploitation and local search abilities. Moreover, EPD is a simple but effective mechanism that can be embedded in different optimizers. Thus, we consider to adopt EPD to enhance the searching ability for feature selection.

Algorithm 2 IBDA

---

1   Define and initialize the related parameters: swarm size N_{swarm}, solution dimension N_{dim}, maximum iteration N_{max_iter}, and fitness function, etc.;

2   **for** $t = 1$ *to* N_{max_iter} **do**

3   Calculate the fitness function values of solutions;

4   Sort these solutions in ascending order according to their fitness function values;

5   Update the food source and enemy, namely, *food_position* and *enemy_position*;

6   Reinitialize the individuals from the last half of ordered swarm by using Algorithm 3;

7   Update w, s, a, c, f, and e;

8   **for** $i = 1$ *to* N_{swarm} **do**

9   Calculate S_i, A_i, C_i, $Food_i$, $Emery_i$ by using Eqs. (2), (3), (4), (5), (6);

10   Update the binary position vector of ith dragonfly of the swarm by using Algorithm 5;

11   **end**

12   **end**

13   Return *food_position*;

14   //*food_position* **is the best solution obtained by the algorithm**

The core idea of EPD is to eliminate the worst solutions, improve the median fitness of the whole swarm, and relocate the removed solutions according to the best solutions. Note that the worst and best solutions are the solutions with worst and best fitness functions. To combine DA with EPD, the dragonfly swarm is divided into two parts according to their fitness function values. Then, half of the solutions in the swarm that with the worst fitness function values are died out and reinitialized based on EPD mechanism. Several EPD mechanisms have been proposed and used in some algorithms, and the popular ones as well as their main principles are introduced as follows.

Basic EPD: For each solution in the worst half of the swarm, EPD selects the best three solutions and generates a new solution by using these selected three solutions. Then, it randomly selects a solution from these four solutions for relocating the original worst solution.

EPD_CM: In this mechanism, the mutation and crossover operators are introduced into the basic EPD. A solution selected by basic EPD is first mutated by using a mutation operator, then the mutated solution is crossed with the original solution by using a crossover operator. Thus, the exploration tendency of the algorithm may be improved by the introduced mutation and crossover operators.

EPD_Tour: In this version, the tournament selection (TS) operator is introduced to select solutions from the best solutions of the swarm. First, N_t solutions are picked out randomly from the best half of the swarm, and the best solution X_{best} is selected among the N_t solutions. Then, X_{best} is operated by using the same mutation and crossover operations as

mentioned in EPD_CM. Compared to EPD_CM, more valuable solutions are retained by the TS operator, thereby making it easier for the algorithm to obtain the optimal solution.

EPD_RWS: Different from the above EPD mechanisms, EPD_RWS utilizes the roulette wheel selection (RWS) operator to select an individual from the best half of the swarm. The basic idea of RWS is that the probability of each individual being selected is proportional to its fitness function value. Then, the selected solution is handled by the same mutation and crossover operations as used in EPD_CM. As can be seen, RWS never ignores any individuals in the swarm, such that more regions of the solution space are explored.

2) EPD WITH LINEAR RANKING SELECTION (LRS)

It has been demonstrated that EPD_Tour and EPD_RWS may achieve better performance than other EPD versions [63], [64]. However, they still have some limitations. In the EPD_Tour, the worst $(N_t - 1)$ individuals in the best half of the swarm will be never selected, which may reduce the population diversity. For EPD_RWS, if the fitness function value of an individual is differed by an order of magnitude with other's fitness function values, then the individual will be rarely selected, which may cause the algorithm to converge prematurely. These conditions above will cause some valuable solutions to be discarded, which makes the algorithm easy to fall into local optimum. Therefore, to overcome these drawbacks, we introduce the linear ranking selection (LRS) method as the selection operator in EPD mechanism and propose a novel EPD_LRS approach to further improve the population diversity.

Algorithm 3 EPD_LRS

---

1   Define and initialize the related parameters: population size N_{swarm}, and the array of solutions sorted by fitness function values in ascending order *Array*, etc.;

2   **for** $k = 0$ *to* $\frac{N_{swarm}}{2}$ **do**

3   Calculate the probability being selected of the kth solutions of the best half by using Eq. (14).

4   **end**

5   **for** $i = \frac{N_{swarm}}{2}$ *to* N_{swarm} **do**

6   Select a solution X^{EPD_LRS} from the best half according to the probabilities.

7   Mutate X^{EPD_LRS} and cross it with X_i by using Algorithm 4;

8   Relocate X_i by using X^{EPD_LRS}.

9   **end**

10   Return *Array*;

11   //*Array* **is the array of relocated solutions**

As shown in algorithm 3, in EPD_LRS, each individual of the best half is first ranked according to the fitness function value. Then, the mechanism determines which solution needs to be operated according to a probability which is designed
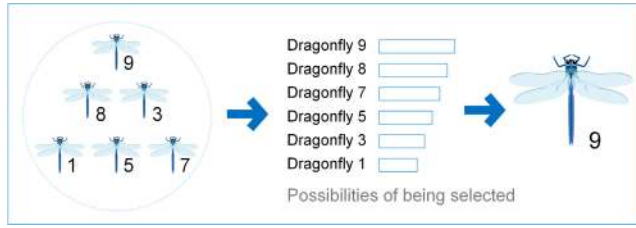
FIGURE 1. The main principle of EPD_LRS.

as follows:

$$P_k = P_{min} + (P_{max} - P_{min}) \frac{k-1}{N_{rank} - 1} \quad (12)$$

where P_{max} and P_{min} are the probabilities for selecting the highest and lowest ranked individuals, respectively. Moreover, P_k is the probability that the kth solution at the intermediate rank will be selected, and N_{rank} is the number of individuals participating in the ranking. The schematic diagram of this method is shown in Fig. 1. P_{max} and P_{min} are designed as follows:

$$P_{max} = \frac{2 \times f_{Fitness_{max}}}{N_{rank} \times (f_{Fitness_{max}} + f_{Fitness_{min}})} \quad (13)$$

$$P_{min} = \frac{2 \times f_{Fitness_{min}}}{N_{rank} \times (f_{Fitness_{max}} + f_{Fitness_{min}})} \quad (14)$$

where $f_{Fitness_{max}}$ and $f_{Fitness_{min}}$ are the fitness function values of the highest and lowest ranked individuals, respectively. By using these methods, the connections between the fitness function values and selection probabilities can be established so that the algorithm may obtain the reasonable operation probability.

Finally, the mutation and crossover operators are used to relocate the obtained solutions so that making them to explore wider areas of solution space. Note that a solution of the formulated feature selection problem may consist of many dimensions, and the mutation and crossover operators are performed on each dimension of the solution. In our mechanism, a widely used mutation operator is adopt and it is defined as follows [63]:

$$x = \begin{cases} 1-x, & N_{rand} \leq \varphi \\ x, & otherwise \end{cases} \quad (15)$$

where x represents a dimension of a solution, N_{rand} expresses a random number within the range of [0,1], and φ indicates the linear mutation rate which is calculated as follows:

$$\varphi = 0.9 - \left(1 + \frac{1-t}{N_{max\_iter} - 1}\right) \quad (16)$$

3) AC OPERATOR
The crossover operator can generate a new solution by crossing the solution generated the EPD mechanism and the original solution generated by DA, thereby increasing the population diversity of the algorithm. However, the probabilities of the two solutions being retained in the existing

crossover operators are equal and constant, which causes the algorithm to be difficult to ensure an excellent convergence rate. To solve this problem, we propose an AC operator to balance the exploration and exploitation performances at different iteration stages of the algorithm, so that facilitating the transiting from exploration to exploitation of the search space. Specifically, if the algorithm is with earlier iterative stage, the AC operation should tend to retain the original solutions generated by DA such that the global exploration can be achieved. If the algorithm is proceeded in the later iteration stage, then more solutions generated by the EPD mechanism should be kept to make the algorithm to have a faster convergence rate. To achieve the abovementioned purpose, we design the AC operator as follows:

$$x = \begin{cases} x^{DA}, & N_{rand} \leq \epsilon \\ x^{EPD}, & otherwise \end{cases} \quad (17)$$

where x^{DA} is a dimension of the solution generated by DA, and x^{EPD} indicates a dimension of the solution generated by EPD, respectively. Moreover, ϵ indicates the bias rate of the crossover operator, which is calculated as follows:

$$\epsilon = (0.5 + \theta) - \frac{2\theta}{1 + e^{5 - \frac{10iter}{max\_iter}}} \quad (18)$$

where θ is the maximum bias of the crossover operator. A typical curve of ϵ changes with iterations is shown in Fig. 2. As can be seen, ϵ can be regarded as a bias threshold to make the crossover operator have different tendencies so that retaining the two solutions at different iterations, thereby balancing the exploitation and exploration capabilities at different iteration stages of the algorithm.

Accordingly, the mutation and crossover mechanisms are shown in Algorithm 4. By using these mechanisms, the population diversity is increased and the exploitation and exploration capabilities are balanced, thereby improving the performance of the proposed IBDA for the formulated feature selection problem.

Algorithm 4 Mutation and Crossover Mechanisms

---
1 Define and initialize the related parameters: the original solution X^{DA}, the solution generated by EPD mechanisms X^{EPD}, and solution dimension N_{dim}, etc.;
2 Calculate φ and ϵ by using Eqs. (16) and (18), respectively;
3 **for** $j = 1$ *to* N_{dim} **do**
4   Mutate the jth dimension of X^{EPD} by using Eq. (15);
5   Reinitialize the jth dimension of X^{EPD} by crossing the jth dimensions of X^{DA} and X^{EPD} according to Eq. (17);
6 **end**
7 Return X_{EPD};
8 *//X_{EPD} is the relocated solutions*

Algorithm 5 Binary Mechanism

---

1   Define and initialize the related parameters: solution from previous iteration X_t, and solution dimension N_{dim}, etc.;
2   **for** $j = 1$ *to* N_{dim} **do**
3   Calculate the step vector of the jth dimension by using Eqs. (7);
4   Calculate of the changing probability of the jth dimension of X_t by using Eq. (19);
5   Update the jth dimension of X_{t+1} by using Eq. (20)
6   **end**
7   Return X_{t+1};
8   *//X_{t+1} is is the updated binary solutions*

---
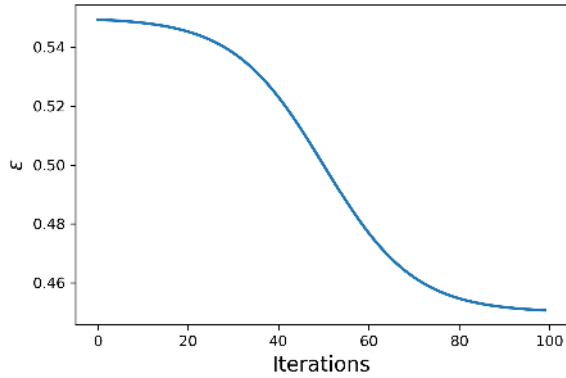
4) BINARY MECHANISM

The solutions in conventional DA are continuous and they can be updated by using the step vectors shown in Eq. (8) directly. However, the solution space of the formulated feature selection problem is discrete, which cannot be handled by conventional DA. Thus, a binary mechanism shown in Algorithm 5 is introduced to map the solutions from the continuous space to discrete space, so that making the algorithm suitable for the feature selection problems.

In this work, the v-shaped transfer function is first utilized to calculate the probability of changing position for each dimension of all solutions. The v-shaped transfer function is described as follows:

$$P(\Delta x) = \left| \frac{\Delta x}{\sqrt{\Delta x^2 + 1}} \right| \tag{19}$$

where Δx means a step vector which is calculated by Eq. (7). The functional relationship between Δx and $P(\Delta x)$ is shown in Fig. 3. As can be seen, the v-shaped transfer function tends to change the variables of search dragonflies more frequently, which boosts exploration in the huge solution space of the formulated feature selection problem. Then, a dimension of the binary solution is updated by using the method as follows:

$$x_{t+1} = \begin{cases} \neg x_t & N_{rand} < P(\Delta x_{t+1}) \\ x_t & otherwise \end{cases} \tag{20}$$

where x_t is a dimension of a binary solution of the tth iteration.

By using the binary mechanism above, the continuous solution space of conventional DA can be effectively transferred to the discrete spaces so that making the algorithm suitable for the feature selection problem.

C. COMPLEXITY ANALYSIS OF IBDA

In this section, the complexity of the proposed IBDA is analyzed. The most time-consuming step in feature selection should be the calculation of fitness function value, which is several orders of magnitude more complex than the other steps. Thus, other calculation steps can be ignored.

FIGURE 3. Sketch map of V-shaped transfer function.

We suppose that the maximum number of iteration and population size are N_{max_iter} and N_{swarm}, respectively, then the complexity of IBDA is $\mathcal{O}(N_{max\_iter} \cdot N_{swarm})$ because the fitness function values are calculated $N_{max\_iter} \cdot N_{swarm}$ times in the algorithm, which is the same with the conventional DA. However, IBDA may consume extra computing time than conventional DA in practical application, and the reason may be that the introduced improved factors lead to additional computing time, which is difficult to be predicted. Thus, to assess the computational complexity of the algorithm more comprehensively, the experiment time of the proposed algorithm is evaluated in Section V.

D. FEATURE SELECTION WITH IBDA

To solve the feature selection problem by utilizing the proposed method, we consider the dragonfly of the swarm as a solution to the problem. Therefore, the dragonfly consists of a one-dimensional vector is actually a solution of the problem, in which the value of each bit is 1 or 0 (1 and 0 indicate that the feature will be chosen or not, respectively). Thus, a dragonfly can be expressed as follows:

$$X = (x_1, x_2, x_3, \ldots, x_{N_{dim}}) \tag{21}$$

TABLE 1. Benchmark datasets.

No.	Dataset	No. of Features	No. of instance
1	Arrhythmia	278	452
2	Breastcancer	10	699
3	BreastEW	30	569
4	Congress	16	435
5	Dermatology	34	366
6	Diabetes	8	768
7	German	24	1000
8	Glass	9	214
9	Heart-StatLog	13	270
10	Hepatitis	19	142
11	Hillvalley	100	606
12	Ionosphere	34	351
13	Krvskp	36	3196
14	Lung	72	326
15	Lung-Cancer	56	32
16	Lymphography	18	148
17	Movementlibras	90	360
18	semeion	265	1593
19	Sonar	60	208
20	Spect	22	267
21	Tic-tac-toe	9	958
22	Vehicule	18	846
23	Vowel	10	901
24	WaveformEW	40	5000
25	WDBC	31	569
26	Wine	13	178
27	Zoo	16	101

TABLE 2. Parameter tuning results of τ and ζ in IBDA.

		$\zeta = 0.1$	$\zeta = 0.2$	$\zeta = 0.3$	$\zeta = 0.4$
$\tau = 0.6$	Accuracy	**0.9758**	0.9750	0.9750	0.9750
	Features	**8.73**	9.93	8.87	9.37
	Fitness	**0.02548**	0.02652	0.02633	0.02642
$\tau = 0.7$	Accuracy	0.9750	0.9750	0.9750	0.9742
	Features	8.27	7.87	9.53	10.50
	Fitness	0.02623	0.02615	0.02645	0.02745
$\tau = 0.8$	Accuracy	0.9750	0.9750	0.9742	0.9750
	Features	7.30	7.97	8.47	10.30
	Fitness	0.02605	0.02617	0.02709	0.02659
$\tau = 0.9$	Accuracy	0.9750	0.9750	0.9750	0.9750
	Features	8.10	7.77	9.80	8.57
	Fitness	0.02620	0.02614	0.02650	0.02628

TABLE 3. Key parameters of different algorithms.

No.	Algorithm	Parameters
1	BPSO	$c1 = 2, c2 = 2$
2	BGWO	$\alpha = [2, 0]$
3	BDA	$w = [0.9, 0.4], s = [0.2, 0], a = [0.2, 0],$ $c = [0.2, 0], f = [0.2, 0], e = [0, 0.1]$
4	BBA	$A = 0.25, Q_{max} = 2, Q_{min} = 0$
5	IBPSO	$c1 = 1.75, c2 = 2$
6	IBDA	$w = [0.6, 0.1], s = [0.2, 0], a = [0.2, 0],$ $c = [0.2, 0], f = [0.2, 0], e = [0, 0.1]$

where N_{dim} represents the number of features. Moreover, the swarm of IBDA is expressed as follows:

$$Swarm = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{N_{swarm}} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{N_{dim}}^1 \\ x_1^2 & x_2^2 & \cdots & x_{N_{dim}}^2 \\ \vdots & \vdots & & \vdots \\ x_1^{N_{swarm}} & x_2^{N_{swarm}} & \cdots & x_{N_{dim}}^{N_{swarm}} \end{bmatrix} \tag{22}$$

V. EXPERIMENTS AND ANALYSIS

In this section, tests are conducted to verify the performance of the proposed IBDA for the feature selection problem. First, the datasets and setups used in the experiments are introduced. Then, the test results obtained by IBDA and several comparison algorithms are presented and analyzed. Finally, the effectiveness of the introduced improved factors are evaluated.

A. DATASETS AND SETUPS

1) BENCHMARK DATASETS

In this work, we use 27 datasets that are selected from the UC Irvine Machine Learning Repository to perform the experiments. The main information of these datasets are shown in Table 1.

2) PARAMETER TUNING

As we mentioned above, IBDA regards all the dragonflies as one sub-swarm, and it simulates the exploration and exploitation processes of the algorithm by adaptively tuning the swarming factors including s, a, c, f, e, and the inertia weight w. However, it is difficult to tune all the parameters since there are six parameters in IBDA. Thus, refer to the experiments in [13], we use two parameters that are τ and ζ, to calculate these parameters as follows:we use two alternative parameters that are τ and ζ to represent the original parameters of IBDA (see Eqs. (23)-(29)), so that tuning the parameters in a reasonable way.

w is calculated as follows:

$$w = \tau - t \times \frac{0.5}{N_{max\_iter}} \tag{23}$$

where τ is a parameter that controls the size of w. Specifically, $w \in [\tau, \tau - 0.5]$.

Moreover, s, a, c, f, and e are calculated as follows:

$$s = 2 \times N_{rand} \times \xi \tag{24}$$
$$a = 2 \times N_{rand} \times \xi \tag{25}$$
$$f = 2 \times N_{rand} \times \xi \tag{26}$$
$$c = 2 \times N_{rand} \times \xi \tag{27}$$
$$e = \xi \tag{28}$$

where ξ is calculated as follows:

$$\xi = \begin{cases} \zeta - t \times \dfrac{\zeta - (-\zeta)}{N_{max\_iter}}, & t \le \dfrac{N_{max\_iter}}{2} \\ 0, & otherwise \end{cases} \tag{29}$$

where ζ is a parameter that controls the size of s, a, c, f, and e.

TABLE 4. Fitness function values obtained by different algorithms.

	BDA		BBA		BGWO		BPSO		IBPSO		IBDA	
	fitness	std	fitness	std	fitness	std	fitness	std	fitness	std	fitness	std
Arrhythmia	0.3236	0.0106	0.3392	0.0060	0.3432	0.0078	0.3405	**0.0055**	0.3190	0.0099	**0.3147**	0.0125
Breastcancer	0.0290	0.0023	0.0274	0.0007	0.0290	0.0031	**0.0272**	**0.0000**	**0.0272**	**0.0000**	**0.0272**	**0.0000**
BreastEW	0.0406	0.0062	0.0401	0.0026	0.0448	0.0055	0.0408	0.0020	0.0394	0.0005	**0.0389**	**0.0000**
Congress	0.0310	0.0033	**0.0251**	**0.0021**	0.0305	0.0060	0.0250	0.0023	0.0264	0.0028	0.0253	0.0022
Dermatology	0.0193	0.0054	0.0128	0.0031	0.0169	0.0042	0.0138	**0.0019**	0.0116	0.0031	**0.0108**	0.0032
Diabets	0.2558	0.0015	0.2557	0.0002	0.2580	0.0026	**0.2557**	**0.0000**	**0.2557**	**0.0000**	**0.2557**	**0.0000**
German	0.2541	0.0076	0.2476	0.0052	0.2552	0.0063	0.2508	**0.0042**	0.2474	0.0056	**0.2446**	0.0059
Glass	0.3151	0.0066	**0.3071**	0.0031	0.3266	0.0107	**0.3071**	**0.0000**	**0.3071**	**0.0000**	**0.3071**	**0.0000**
Heart-StatLog	0.1557	0.0186	**0.1416**	0.0009	0.1590	0.0190	0.1417	0.0010	**0.1416**	**0.0000**	**0.1416**	**0.0000**
Hepatitis	0.2811	0.0152	0.2680	0.0123	0.2921	0.0133	0.2735	**0.0053**	0.2667	0.0065	0.2697	0.0093
Hillvalley	0.3947	0.0146	0.3951	0.0057	0.3937	0.0064	0.4005	**0.0053**	0.3791	0.0100	**0.3702**	0.0139
Ionosphere	0.1009	0.0127	0.1044	0.0106	0.1283	0.0113	0.1133	0.0090	0.0919	**0.0084**	**0.0840**	0.0092
Krvskp	0.0353	0.0099	0.0302	0.0028	0.0382	0.0061	0.0354	0.0030	0.0259	0.0023	**0.0248**	**0.0015**
Lung	0.1007	0.0210	0.0989	0.0131	0.1117	0.0154	0.1062	**0.0086**	0.0799	0.0129	**0.0619**	0.0187
Lung-Cancer	0.0270	0.0165	0.0278	0.0029	0.0325	0.0116	0.0285	0.0022	0.0266	**0.0003**	**0.0259**	0.0032
Lymphography	0.5800	0.0122	0.5602	0.0111	0.5820	0.0147	0.5646	0.0086	**0.5530**	**0.0038**	0.5650	0.0109
Movementlibras	0.1924	0.0079	0.1829	**0.0036**	0.1873	0.0058	0.1887	0.0039	0.1747	0.0081	**0.1691**	0.0076
semeion	0.0283	0.0045	0.0279	0.0013	0.0284	0.0021	0.0290	**0.0012**	0.0251	0.0023	**0.0231**	0.0027
Sonar	0.1015	0.0187	0.0893	0.0085	0.1048	0.0125	0.1021	0.0077	**0.0780**	**0.0044**	0.0815	0.0127
Spect	0.2766	0.0123	0.2637	0.0085	0.2815	0.0118	0.2693	**0.0069**	0.2599	0.0079	0.2675	0.0107
Tic-tac-toe	0.1920	0.0206	**0.1564**	**0.0000**	**0.1564**	**0.0000**	**0.1564**	**0.0000**	**0.1564**	**0.0000**	**0.1564**	**0.0000**
Vehicule	0.2917	0.0072	0.2709	**0.0030**	0.2826	0.0100	0.2732	0.0094	**0.2696**	0.0060	0.2700	0.0040
Vowel	0.0623	0.0032	**0.0590**	**0.0000**	0.0599	0.0035	**0.0590**	**0.0000**	**0.0590**	**0.0000**	**0.0590**	**0.0000**
WaveformEW	0.1783	0.0084	0.1687	0.0040	0.1731	0.0042	0.1732	**0.0032**	0.1621	0.0060	0.1625	0.0058
WDBC	0.0438	0.0056	0.0403	0.0031	0.0511	0.0086	0.0421	0.0023	0.0395	0.0009	**0.0389**	**0.0000**
Wine	0.0538	0.0064	**0.0478**	**0.0000**	0.0579	0.0160	0.0482	0.0023	0.0483	0.0012	0.0482	0.0028
Zoo	0.0859	0.0112	0.0623	0.0087	0.0826	0.0127	0.0625	0.0065	**0.0630**	**0.0043**	0.0634	0.0089

TABLE 5. CPU time occupied by different algorithms.

	BDA		BBA		BGWO		BPSO		IBPSO		IBDA	
	time	std	time	std	time	std	time	std	time	std	time	std
Arrhythmia	**168.4**	30.06	216.7	14.49	295.9	15.24	196.7	**3.50**	212.7	14.53	192.3	11.36
Breastcancer	60.2	2.52	60.6	1.94	62.0	2.03	**52.4**	1.75	57.1	0.88	57.1	**0.41**
BreastEW	71.2	3.62	73.0	1.77	77.5	3.13	**65.4**	1.51	69.4	0.90	68.1	**0.65**
Congress	61.8	2.90	64.1	1.90	61.0	3.81	**57.9**	1.87	60.0	1.46	61.4	**0.54**
Dermatology	55.6	2.47	**35.9**	**0.26**	67.3	2.39	52.8	1.13	61.0	0.59	59.0	0.46
Diabets	55.8	4.14	57.9	2.88	53.7	3.93	**49.7**	2.62	53.9	3.02	57.7	**0.43**
German	71.8	7.39	71.5	4.86	74.9	6.49	**62.0**	2.98	77.9	3.87	71.4	**2.82**
Glass	45.1	2.12	44.6	1.24	44.4	1.75	**38.8**	1.76	43.7	1.49	48.0	**0.32**
Heart-StatLog	50.5	1.94	49.4	1.45	50.5	2.41	**43.3**	1.26	49.3	0.42	52.7	**0.27**
Hepatitis	43.4	1.73	42.9	1.93	43.7	1.73	**37.8**	1.43	39.7	0.57	46.3	**0.33**
Hillvalley	**84.7**	13.06	96.7	3.25	120.6	4.93	86.9	**2.48**	99.3	3.14	96.1	5.06
Ionosphere	60.1	3.95	61.7	2.24	64.4	3.55	**55.8**	1.64	57.2	**0.88**	63.6	0.89
Krvskp	**640.6**	129.55	792.2	71.98	1068.0	237.42	702.4	**34.28**	751.0	70.19	659.8	67.36
Lung	63.6	3.41	62.5	2.00	81.0	2.20	**58.0**	2.45	63.8	0.90	69.6	**0.74**
Lung-Cancer	42.9	1.24	41.3	1.88	43.8	2.43	**36.8**	1.89	39.6	0.56	45.0	**0.26**
Lymphography	44.5	1.76	43.9	1.73	44.6	2.46	**38.3**	1.16	47.9	0.67	47.0	**0.30**
Movementlibras	76.9	7.26	83.7	3.45	99.8	5.15	**75.5**	2.13	77.9	**0.82**	86.6	1.70
semeion	**926.2**	286.17	1511.8	83.63	1865.5	219.60	1355.1	82.85	1433.9	**52.70**	1801.7	115.83
Sonar	55.4	3.07	56.5	1.83	61.3	2.99	**50.1**	1.50	53.1	0.60	59.3	**0.55**
Spect	46.4	2.32	46.2	1.63	48.6	3.33	40.8	1.67	**39.3**	0.61	49.3	0.45
tic-tac-toe	58.3	8.64	59.7	4.97	66.2	3.76	**51.0**	0.81	53.5	6.56	82.6	7.42
Vehicule	78.9	3.23	79.3	3.18	93.1	5.69	**69.9**	3.18	77.5	**1.11**	83.2	1.61
Vowel	68.6	3.32	68.6	2.71	82.9	2.92	**60.7**	3.23	72.5	2.61	72.8	**0.35**
WaveformEW	**2298.0**	487.63	2738.1	210.41	3732.1	380.44	2561.7	109.33	2933.1	**97.66**	2593.5	200.54
WDBC	69.3	4.04	73.9	1.31	76.9	2.51	65.6	1.63	**59.7**	0.84	68.0	**0.59**
Wine	42.7	1.40	46.1	1.01	46.9	0.84	**40.6**	0.92	44.6	1.11	46.5	**0.26**
Zoo	**34.6**	0.25	44.8	0.98	45.8	1.00	39.6	0.93	41.3	0.80	45.0	**0.24**

Theoretically, according to no-free-lunch theory, we need to tune τ and ζ for different optimization problems separately to achieve the best performance for each problem [65], [66].

In this work, each dataset comes from a practical optimization problem, which means that it is better to tune the main parameters for each dataset. However, this will be huge works
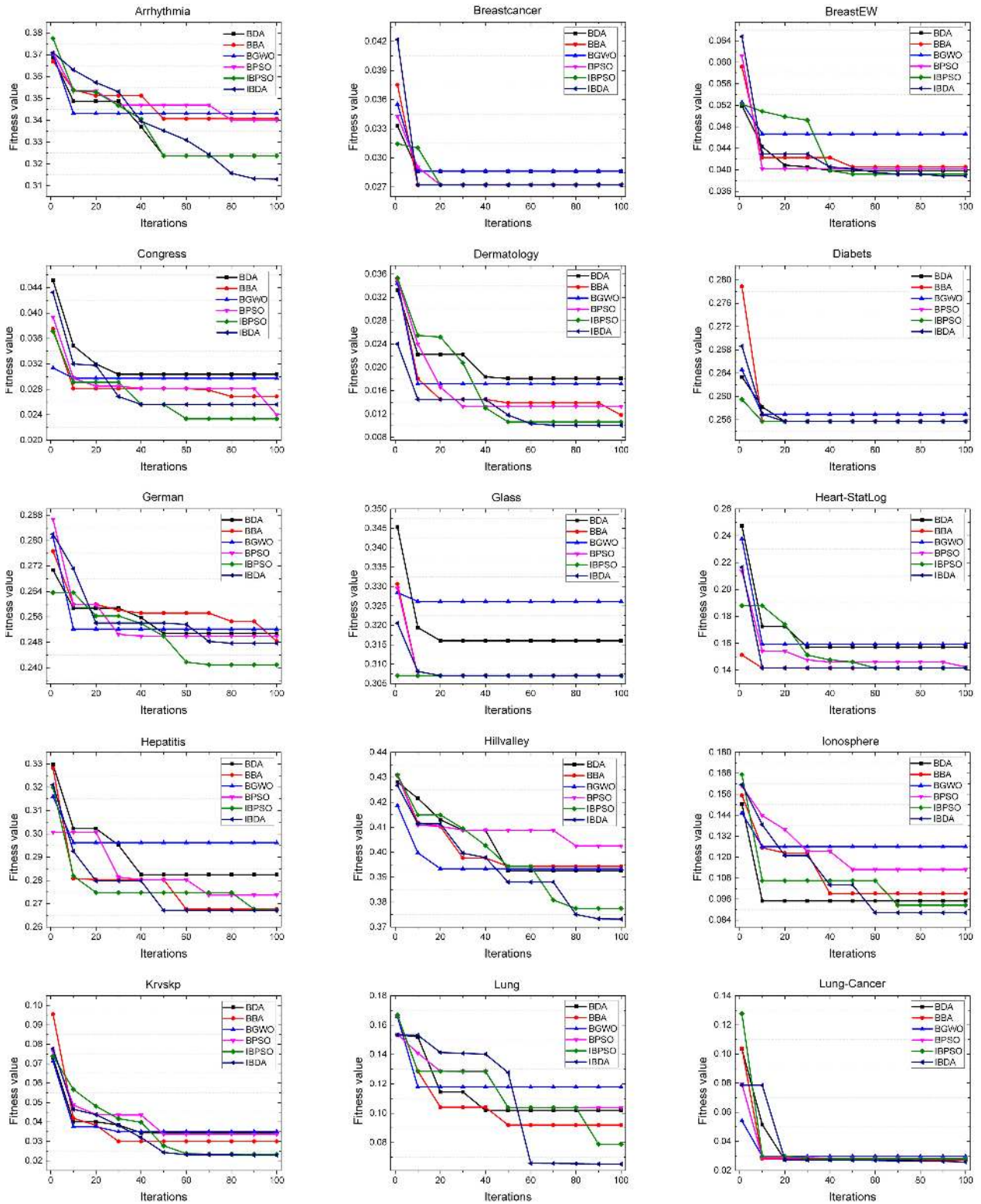
FIGURE 4. Convergence rates obtained by different algorithms (Part 1).

since there are 27 datasets that need to be tuned in our work. For the sake of simplicity, refer to [67], we select one dataset with the median dimension size, namely Lung-Cancer, to tune the key parameters of the proposed IBDA.
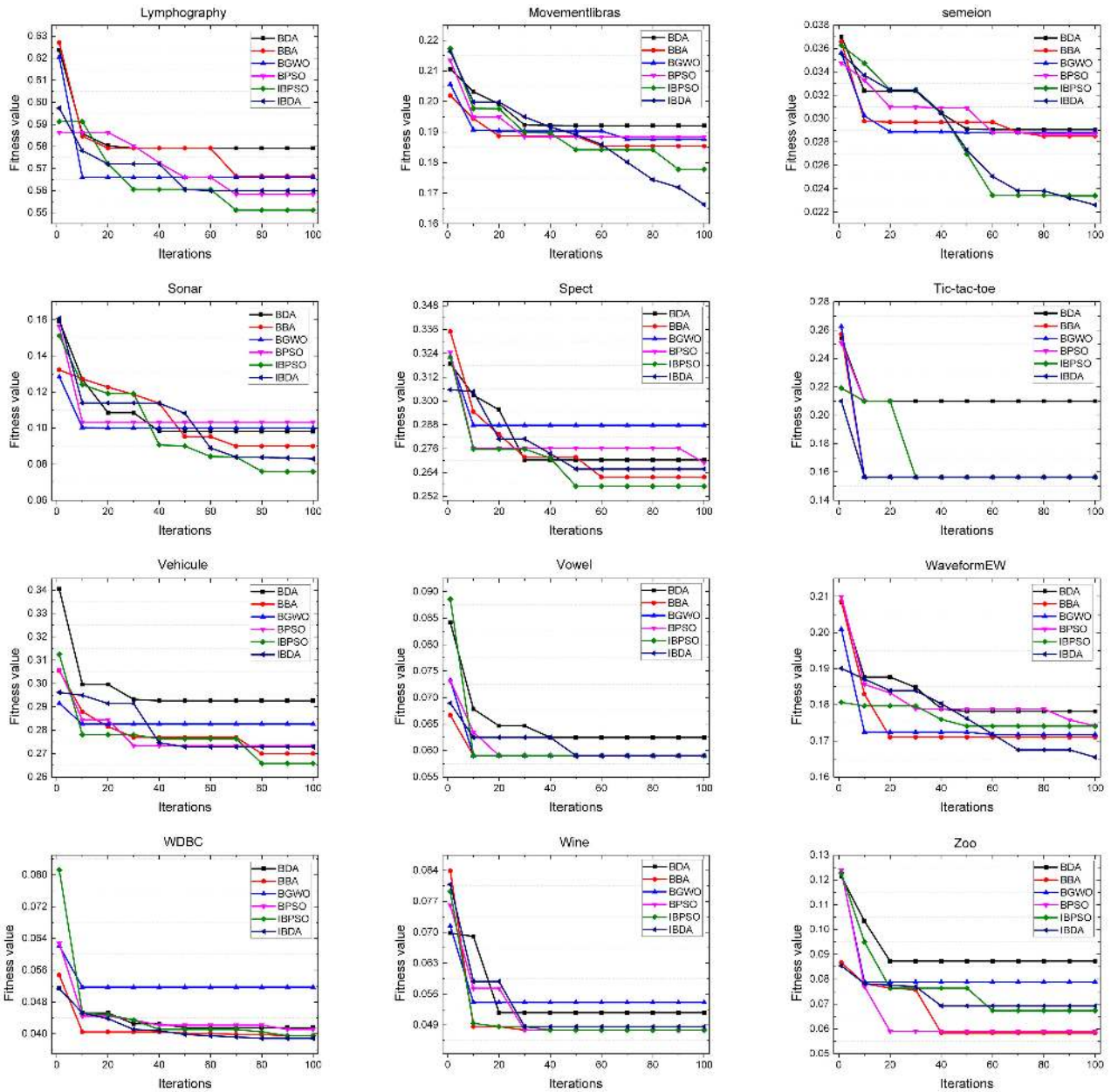
FIGURE 5. Convergence rates obtained by different algorithms (Part 2).

Accordingly, we jointly tune τ and ζ by using the simple generate-evaluate methods [68]. Specifically, we use the proposed IBDA with different combinations of τ and ζ to solve the feature selection problem on Lung-Cancer dataset. Moreover, τ ranges from 0.9 to 0.5 with a step size of 0.1, and ζ is from 0.4 to 0.1 with a step size of 0.1. Thus, there are 16 different combinations of these parameters. Moreover, each combination is independently run for 30 times, and the average results are recorded and shown in Table 2. According to the results, when $\tau = 0.6$ and $\zeta = 0.1$, which means that $w \in [0.6, 0.1]$, $s \in [0.2, 0]$, $a \in [0.2, 0]$, $c \in [0.2, 0]$, $f \in [0.2, 0]$ and $e \in [0.1, 0]$, IBDA achieves the best

optimization results. Thus, we use these parameter values for all the datasets.

3) EXPERIMENT SETUPS
The CPU of the computer used for the experiments is Intel(R) Xeon(R) E5-2630 v4 and the RAM is 32 GB. We implement the experiments by employing Python 3.7 and the KNN (k = 5) based on Euclidean distance measurement is utilized. Moreover, α and β in the fitness function are set to 0.99 and 0.01, respectively.

In this paper, the BPSO [69], BGWO [12], BDA [70], BBA [9] and improved BPSO (IBPSO) [53] are introduced as

TABLE 6. Classification accuracies achieved by different algorithms.

	BDA		BBA		BGWO		BPSO		IBPSO		IBDA	
	acc	std	acc	std	acc	std	acc	std	acc	std	acc	std
Arrhythmia	0.6761	0.0095	0.6622	**0.0045**	0.6607	0.0083	0.6609	0.0069	0.6826	0.0100	**0.6857**	0.0126
Breastcancer	0.9764	0.0038	0.9784	0.0000	0.9771	0.0066	0.9786	0.0000	0.9786	0.0000	0.9786	0.0000
BreastEW	0.9605	0.0052	**0.9614**	0.0017	0.9593	0.0056	0.9611	0.0000	**0.9614**	**0.0000**	**0.9614**	**0.0000**
Congress	0.9725	0.0055	**0.9793**	0.0045	0.9745	0.0046	0.9789	0.0045	0.9773	0.0025	0.9784	0.0048
Dermatology	0.9851	0.0071	0.9924	0.0034	0.9897	**0.0029**	0.9919	0.0040	**0.9938**	0.0030	**0.9938**	0.0031
Diabets	**0.7468**	0.0095	**0.7468**	0.0034	0.7456	0.0080	**0.7468**	**0.0000**	**0.7468**	**0.0000**	**0.7468**	**0.0000**
German	0.7478	0.0079	0.7556	0.0070	0.7495	0.0083	0.7521	0.0064	0.7551	**0.0051**	**0.7574**	0.0065
Glass	0.6868	0.0124	**0.6955**	0.0068	0.6755	0.0146	**0.6955**	**0.0000**	**0.6955**	**0.0000**	**0.6955**	**0.0000**
Heart-StatLog	0.8456	0.0174	**0.8593**	**0.0000**	0.8448	0.0189	**0.8593**	**0.0000**	**0.8593**	**0.0000**	**0.8593**	**0.0000**
Hepatitis	0.7193	0.0175	0.7327	0.0142	0.7107	0.0129	0.7273	**0.0053**	**0.7340**	0.0063	0.7307	0.0080
Hillvalley	0.6044	0.0113	0.6059	0.0058	0.6093	0.0073	0.6003	**0.0057**	0.6215	0.0098	**0.6298**	0.0126
Ionosphere	0.9000	0.0096	0.8972	0.0114	0.8753	0.0114	0.8886	0.0084	0.9100	**0.0081**	**0.9172**	0.0095
Krvskp	0.9688	0.0102	0.9751	0.0041	0.9691	0.0069	0.9696	0.0043	0.9799	**0.0033**	**0.9801**	0.0050
Lung	0.9013	0.0193	0.9050	0.0134	0.8938	0.0171	0.8975	**0.0083**	0.9238	0.0131	**0.9413**	0.0213
Lung-Cancer	**0.9750**	0.0111	**0.9750**	**0.0000**	0.9725	0.0094	**0.9750**	**0.0000**	**0.9750**	**0.0000**	**0.9750**	**0.0000**
Lymphography	0.4187	0.0124	0.4387	0.0122	0.4180	0.0153	0.4340	0.0091	**0.4453**	0.0027	0.4333	0.0112
Movementlibras	0.8086	0.0061	0.8200	0.0053	0.8178	0.0051	0.8142	**0.0047**	0.8283	0.0082	**0.8331**	0.0091
semeion	0.9756	0.0042	0.9768	0.0047	0.9788	**0.0017**	0.9755	0.0045	0.9796	0.0024	**0.9813**	0.0026
Sonar	0.9010	0.0176	0.9148	0.0081	0.9005	0.0102	0.9014	0.0079	**0.9262**	0.0049	0.9210	0.0135
Spect	0.7248	0.0135	0.7385	0.0087	0.7222	0.0130	0.7330	**0.0063**	**0.7422**	0.0078	0.7352	0.0099
Tic-tac-toe	0.8143	0.0288	**0.8521**	**0.0000**	**0.8521**	**0.0000**	**0.8521**	**0.0000**	**0.8521**	**0.0000**	**0.8521**	**0.0000**
Vehicule	0.7098	0.0084	0.7315	**0.0034**	0.7215	0.0098	0.7298	0.0084	**0.7335**	0.0062	0.7326	0.0054
Vowel	0.9452	0.0043	**0.9495**	**0.0000**	0.9488	0.0046	**0.9495**	**0.0000**	**0.9495**	**0.0000**	**0.9495**	**0.0000**
WaveformEW	0.8246	0.0083	0.8349	0.0050	0.8329	0.0047	0.8307	**0.0042**	0.8408	0.0069	**0.8417**	0.0060
WDBC	0.9574	0.0029	**0.9614**	**0.0000**	0.9530	0.0073	0.9604	0.0032	0.9612	0.0005	**0.9614**	**0.0000**
Wine	0.9489	0.0100	**0.9556**	**0.0000**	0.9467	0.0156	**0.9556**	**0.0000**	0.9550	0.0017	**0.9556**	0.0081
Zoo	0.9182	0.0149	**0.9418**	0.0101	0.9236	0.0114	**0.9418**	0.0073	0.9409	**0.0045**	0.9409	0.0112

TABLE 7. Number of selected features obtained by different algorithms.

	BDA		BBA		BGWO		BPSO		IBPSO		IBDA	
	atts	std	atts	std	atts	std	atts	std	atts	std	atts	std
Arrhythmia	**81.70**	22.01	132.70	7.73	201.10	11.78	133.40	7.43	132.70	**7.06**	98.30	16.01
Breastcancer	**5.70**	0.41	6.00	**0.00**	6.40	0.88	6.00	**0.00**	6.00	**0.00**	6.00	**0.00**
BreastEW	4.60	2.03	5.60	1.20	13.60	2.12	6.60	0.94	3.60	1.62	**2.00**	**0.00**
Congress	**6.10**	1.18	7.40	**0.80**	8.40	1.65	6.50	1.12	6.20	2.18	6.20	0.82
Dermatology	**15.70**	2.31	18.10	3.66	23.00	2.22	19.50	2.19	18.50	2.46	15.90	**1.54**
Diabets	4.10	1.03	**4.00**	0.37	4.90	0.92	**4.00**	**0.00**	**4.00**	**0.00**	**4.00**	**0.00**
German	**10.60**	2.81	13.50	2.15	17.30	2.45	12.80	2.14	11.90	3.14	**10.60**	**1.60**
Glass	4.50	0.74	5.00	0.44	4.80	0.91	5.00	0.00	5.00	0.00	5.00	0.00
Heart-StatLog	3.60	1.02	**3.00**	0.26	7.00	1.11	3.10	0.26	**3.00**	**0.00**	**3.00**	**0.00**
Hepatitis	6.10	1.92	6.40	1.64	10.70	1.82	6.70	1.39	6.40	**0.66**	5.90	1.25
Hillvalley	**31.20**	11.97	49.10	4.69	69.90	**4.47**	48.30	4.69	44.00	6.29	37.00	8.37
Ionosphere	**6.60**	3.22	9.00	1.66	16.30	2.60	10.40	1.75	9.40	2.54	6.80	**1.63**
Krvskp	**15.70**	2.62	19.80	3.48	27.20	**2.46**	18.80	2.58	21.60	3.56	18.60	2.53
Lung	**94.80**	31.25	157.50	11.17	211.60	14.93	154.30	**6.59**	143.30	8.20	122.00	19.53
Lung-Cancer	12.60	5.50	17.10	3.40	29.70	3.66	20.80	2.60	10.60	**1.69**	**6.50**	3.68
Lymphography	8.10	1.82	8.00	1.39	10.50	1.62	7.70	**1.31**	7.00	2.05	7.20	1.84
Movementlibras	**26.00**	8.19	42.30	5.01	62.50	4.35	42.40	4.69	42.60	**2.73**	34.10	4.24
semeion	**108.70**	22.37	129.80	6.92	197.80	9.98	125.30	7.45	131.90	**5.79**	121.40	13.82
Sonar	20.90	4.36	29.20	4.52	37.70	3.51	27.00	2.62	29.50	3.01	**19.30**	3.54
Spect	**9.10**	2.40	10.60	1.75	14.20	1.50	10.90	1.79	10.30	**1.42**	11.70	1.85
Tic-tac-toe	**7.30**	1.86	9.00	**0.00**	9.00	**0.00**	9.00	**0.00**	9.00	**0.00**	9.00	**0.00**
Vehicule	**7.90**	1.63	9.20	**1.02**	12.50	1.19	10.20	2.05	10.50	1.02	9.50	1.42
Vowel	**8.00**	0.62	9.00	**0.00**	9.20	0.92	9.00	**0.00**	9.00	**0.00**	9.00	**0.00**
WaveformEW	18.67	3.13	21.33	2.35	30.67	**2.09**	22.67	2.51	**18.00**	3.00	23.33	2.47
WDBC	5.00	2.56	6.50	1.12	14.10	2.26	8.90	1.35	3.50	1.36	**2.00**	**0.00**
Wine	**4.10**	1.35	5.00	**0.00**	6.60	1.07	5.50	0.99	4.90	0.70	5.40	1.15
Zoo	7.90	1.88	7.50	0.79	11.20	1.49	7.90	0.67	**7.20**	**0.60**	7.80	1.20

the comparison algorithms. Note that BDA is a binary version of conventional DA and it has the same binary mechanism as IBDA. Table 3 shows the key parameter selections of these algorithms and these parameters are assigned as the values that perform well in the literature for feature selection such that the corresponding algorithms can make a reasonable

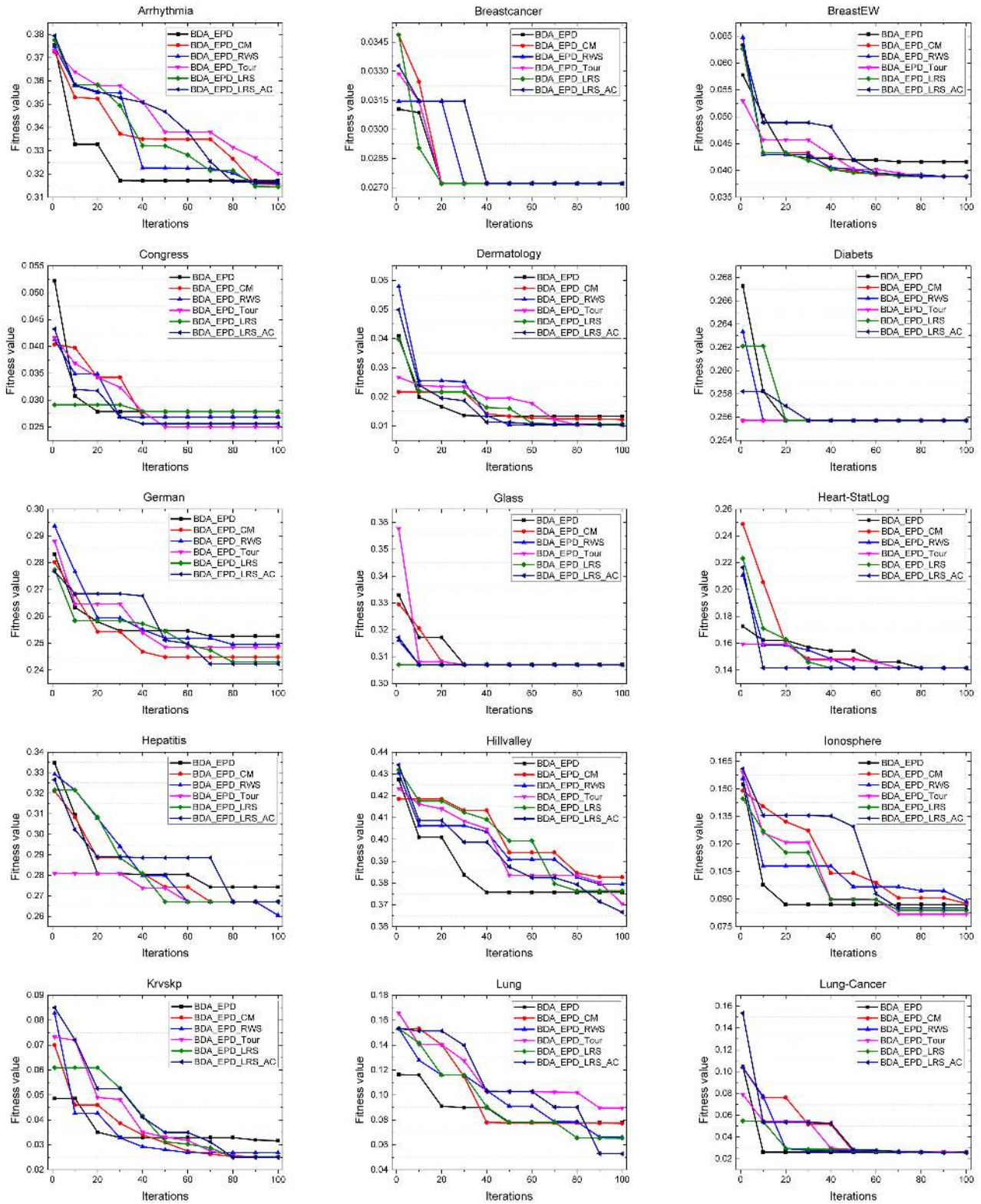**FIGURE 6.** Convergence rates obtained by different improved factors (Part 1).

comparison. Moreover, the proposed IBDA and these comparison algorithms are both metaheuristics algorithms, and they are directly affected by the population size and number

of iterations. Thus, to ensure the fairness of comparison, the same swarm size and number of iterations are utilized for each algorithm. In this work, they are set as 24 and 100,
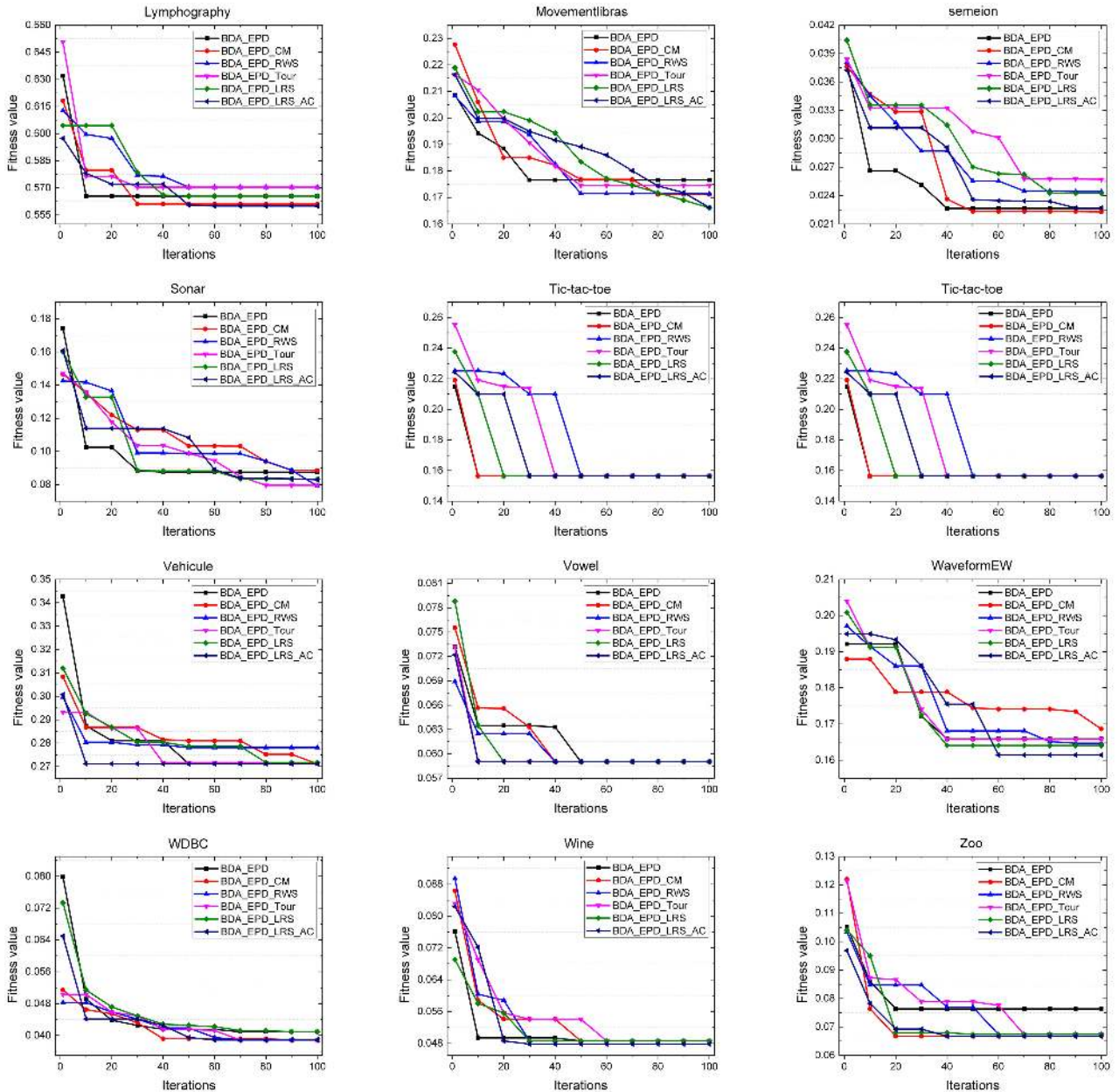
FIGURE 7. Convergence rates obtained by different improved factors (Part 2).

respectively. Furthermore, as indicated by the central limit theorem, each algorithm is independently run for 30 times in these selected datasets to avoid the random bias of the experiment. In addition, 80% of the instances are used for training, and the rest instances are used for testing, which is a common way employed by several previous works.

B. FEATURE SELECTION RESULTS
In this section, the feature selection results in terms of the fitness function value, convergence rate, accuracy and CPU time obtained by different algorithms are presented. Moreover, the best values obtained by a certain approach are highlighted in bold font.

TABLE 8. Introductions of different improved factors.

No.	Algorithm	Improved factors
1	BDA_EPD	EPD strategy
2	BDA_EPD_CM	EPD_CM strategy
3	BDA_EPD_RWS	EPD_RWS strategy
4	BDA_EPD_Tour	EPD_Tour strategy (N_t=3)
5	BDA_EPD_LRS	EPD_LRS strategy
6	BDA_EPD_LRS_AC	EPD_LRS strategy with AC operator

Table 4 shows the numerical statistical results of the average fitness function values and standard deviations (Stds) of different algorithms for each dataset. As can be seen,

TABLE 9. Fitness function values obtained by different improved factors.

	BDA_EPD		BDA_EPD_CM		BDA_EPD_RWS		BDA_EPD_Tour		BDA_EPD_LRS		BDA_EPD_LRS_AC	
	fitness	std	fitness	std	fitness	std	fitness	std	fitness	std	fitness	std
Arrhythmia	0.3161	0.0127	0.3160	0.0092	0.3162	0.0116	0.3186	0.0135	0.3155	**0.0086**	**0.3147**	0.0126
Breastcancer	0.0280	0.0011	**0.0272**	**0.0000**	**0.0272**	**0.0000**	**0.0272**	**0.0000**	**0.0272**	**0.0000**	**0.0272**	**0.0000**
BreastEW	0.0409	0.0034	**0.0389**	0.0026	**0.0389**	**0.0000**	**0.0389**	**0.0000**	**0.0389**	**0.0000**	**0.0389**	**0.0000**
Congress	0.0276	0.0025	0.0263	0.0024	0.0264	0.0018	**0.0250**	0.0023	0.0267	0.0024	0.0253	0.0022
Dermatology	0.0140	**0.0028**	0.0118	**0.0028**	0.0103	0.0034	0.0116	0.0030	0.0112	**0.0028**	0.0108	0.0032
Diabets	**0.2557**	**0.0000**	**0.2557**	**0.0000**	**0.2557**	**0.0000**	**0.2557**	**0.0000**	**0.2557**	**0.0000**	**0.2557**	**0.0000**
German	0.2529	**0.0045**	0.2453	0.0063	0.2491	0.0066	0.2487	0.0061	0.2450	0.0058	**0.2446**	0.0060
Glass	0.3080	0.0028	**0.3071**	**0.0000**	**0.3071**	**0.0000**	**0.3071**	**0.0000**	**0.3071**	**0.0000**	**0.3071**	**0.0000**
Heart-StatLog	0.1422	0.0034	**0.1416**	**0.0000**	0.1417	0.0010	**0.1416**	**0.0000**	**0.1416**	**0.0000**	**0.1416**	**0.0000**
Hepatitis	0.2719	0.0119	0.2669	**0.0065**	**0.2663**	0.0083	0.2695	0.0075	**0.2663**	0.0068	0.2697	0.0094
Hillvalley	0.3751	**0.0108**	0.3837	0.0159	0.3788	0.0112	0.3719	0.0147	0.3735	0.0139	**0.3702**	0.0140
Ionosphere	0.0851	0.0097	0.0902	0.0115	0.0896	**0.0093**	0.0848	0.0118	**0.0830**	0.0122	0.0840	**0.0093**
Krvskp	0.0311	0.0044	0.0249	0.0019	0.0267	0.0024	**0.0247**	0.0026	0.0252	0.0018	0.0248	**0.0015**
Lung	0.0747	0.0183	0.0775	0.0210	0.0680	0.0206	0.0874	**0.0162**	0.0691	0.0218	**0.0619**	0.0189
Lung-Cancer	0.0262	0.0032	0.0259	**0.0020**	0.0259	0.0024	0.0258	0.0028	**0.0257**	0.0023	0.0259	0.0032
Lymphography	0.5652	0.0119	**0.5617**	0.0128	0.5652	**0.0091**	0.5659	0.0116	0.5630	0.0107	0.5650	0.0110
Movementlibras	0.1755	**0.0058**	0.1711	0.0075	0.1701	0.0074	0.1746	0.0069	**0.1677**	0.0071	0.1691	0.0077
semeion	0.0227	0.0031	**0.0222**	0.0029	0.0243	0.0029	0.0249	**0.0026**	0.0239	0.0033	0.0231	0.0027
Sonar	0.0863	0.0134	0.0872	0.0152	0.0790	0.0142	**0.0778**	0.0151	0.0837	**0.0118**	0.0815	0.0128
Spect	0.2622	0.0093	**0.2619**	**0.0091**	0.2649	0.0118	0.2635	0.0107	0.2674	0.0095	0.2675	0.0108
tic-tac-toe	0.1671	0.0180	0.1618	0.0130	**0.1564**	**0.0000**	**0.1564**	**0.0000**	**0.1564**	**0.0000**	**0.1564**	**0.0000**
Vehicule	0.2718	0.0088	0.2720	0.0043	0.2741	0.0110	0.2719	0.0108	0.2717	0.0106	0.2700	0.0040
Vowel	0.0598	0.0010	**0.0590**	**0.0000**	**0.0590**	**0.0000**	**0.0590**	**0.0000**	**0.0590**	**0.0000**	**0.0590**	**0.0000**
WaveformEW	0.1655	0.0049	0.1684	**0.0037**	0.1648	0.0044	0.1679	0.0043	0.1643	0.0046	**0.1625**	0.0059
WDBC	0.0405	0.0044	**0.0389**	0.0027	0.0391	0.0044	**0.0389**	**0.0000**	**0.0389**	**0.0000**	**0.0389**	**0.0000**
Wine	0.0489	0.0041	0.0492	**0.0021**	0.0492	**0.0021**	0.0488	0.0024	0.0485	0.0035	**0.0482**	0.0029
Zoo	0.0746	0.0101	0.0647	0.0087	0.0661	0.0098	0.0642	**0.0068**	0.0670	0.0087	**0.0634**	0.0090

TABLE 10. CPU time occupied by different improved factors.

Datasets	BDA_EPD		BDA_EPD_CM		BDA_EPD_RWS		BDA_EPD_Tour		BDA_EPD_LRS		BDA_EPD_LRS_AC	
	time	std	time	std	time	std	time	std	time	std	time	std
Arrhythmia	**189.6**	22.91	260.1	13.25	195.0	**11.17**	260.8	12.42	234.1	16.67	192.3	11.36
Breastcancer	67.2	3.83	**56.6**	0.49	57.1	0.46	56.7	0.47	57.2	0.55	57.1	**0.41**
BreastEW	**65.9**	1.39	67.9	0.54	67.8	0.49	68.1	0.82	68.3	**0.48**	68.1	0.65
Congress	**59.5**	0.91	61.0	0.51	60.9	**0.44**	60.7	0.51	62.1	0.85	61.4	0.54
Dermatology	59.0	1.18	**58.5**	0.65	58.7	0.94	59.1	0.66	59.4	0.76	59.0	**0.46**
Diabets	**56.9**	2.21	57.3	**0.31**	57.5	**0.31**	57.6	0.35	57.7	0.49	57.7	0.43
German	**71.3**	5.27	72.3	4.12	73.3	3.97	72.6	4.02	72.3	3.26	71.4	**2.82**
Glass	47.4	0.29	47.5	**0.18**	47.8	0.29	47.8	0.23	48.2	0.33	48.0	0.32
Heart-StatLog	**51.5**	0.43	51.9	0.36	52.2	0.35	52.5	0.32	52.1	**0.27**	52.7	**0.27**
Hepatitis	**45.5**	0.54	46.1	0.50	46.4	0.29	46.3	**0.27**	46.5	0.29	46.3	0.33
Hillvalley	**81.7**	7.10	96.5	5.31	95.9	**4.69**	94.9	5.55	97.2	4.72	96.1	5.06
Ionosphere	**59.6**	1.51	63.7	0.95	64.8	0.99	63.4	**0.86**	63.9	**0.86**	63.6	0.89
Krvskp	670.1	105.81	690.7	79.34	673.6	72.36	693.3	78.78	680.7	**60.34**	**659.8**	67.36
Lung	**65.8**	1.13	69.5	**0.43**	69.5	0.75	69.6	0.67	69.8	0.47	69.6	0.74
Lung-Cancer	**44.3**	0.26	44.7	0.22	44.9	**0.20**	44.9	0.28	45.1	0.39	45.0	0.26
Lymphography	**46.2**	0.41	46.5	0.29	46.9	0.31	46.7	**0.24**	47.1	0.35	47.0	0.30
Movementlibras	**79.8**	3.53	87.6	2.29	88.0	**1.46**	87.4	2.73	88.1	3.08	86.6	1.70
semeion	**1726.5**	145.60	1762.2	78.64	1769.1	110.57	1784.9	131.10	1762.1	**75.66**	1801.7	115.83
Sonar	**58.4**	0.76	59.2	0.70	59.5	0.57	59.3	0.61	59.9	0.68	59.3	**0.55**
Spect	**48.3**	0.76	49.1	**0.38**	49.5	0.48	49.2	0.47	49.4	0.43	49.3	0.45
Tic-tac-toe	**81.4**	8.27	84.1	**6.55**	88.0	6.63	84.2	7.28	86.1	6.92	82.6	7.42
Vehicule	**81.3**	2.07	82.6	1.66	83.4	1.88	82.5	2.20	83.0	2.15	83.2	**1.61**
Vowel	73.4	0.74	**72.0**	0.50	72.4	0.52	72.6	0.72	73.0	0.50	72.8	**0.35**
WaveformEW	2463.3	**140.96**	2675.5	214.69	2545.1	227.31	2574.6	257.84	2636.7	254.53	2593.5	200.54
WDBC	**65.4**	1.09	68.0	0.60	65.4	1.09	68.2	0.61	68.2	0.61	68.0	**0.59**
Wine	**45.8**	0.31	46.0	**0.20**	46.3	0.21	46.4	0.27	46.4	0.30	46.5	0.26
Zoo	**44.7**	0.23	44.8	0.30	44.9	**0.17**	44.9	0.72	45.3	0.74	45.0	0.24

BPSO and BBA achieve better fitness function values than BDA for the datasets with less number of features. However, BDA has better performance for the datasets with larger numbers of features. This demonstrates our conjecture that BDA may have a good exploration ability but lacks the exploitation performance. Thus, by introducing the improved

TABLE 11. Classification accuracies achieved by different improved factors.

	BDA_EPD		BDA_EPD_CM		BDA_EPD_RWS		BDA_EPD_Tour		BDA_EPD_LRS		BDA_EPD_LRS_AC	
	acc	std	acc	std	acc	std	acc	std	acc	std	acc	std
Arrhythmia	0.6835	0.0114	0.6843	0.0093	0.6841	0.0122	0.6820	0.0137	0.6848	**0.0075**	**0.6857**	0.0126
Breastcancer	0.9776	0.0037	**0.9786**	**0.0000**	**0.9786**	**0.0000**	**0.9786**	**0.0000**	**0.9786**	**0.0000**	**0.9786**	**0.0000**
BreastEW	0.9600	0.0030	0.9614	0.0025	**0.9614**	**0.0000**	**0.9614**	**0.0000**	**0.9614**	**0.0000**	**0.9614**	**0.0000**
Congress	0.9761	0.0055	0.9775	**0.0048**	0.9770	0.0049	**0.9786**	0.0049	0.9770	0.0050	0.9784	0.0048
Dermatology	0.9908	0.0042	0.9927	0.0036	**0.9943**	0.0037	0.9927	**0.0026**	0.9935	0.0030	0.9938	0.0031
Diabets	**0.7468**	**0.0000**	**0.7468**	**0.0000**	**0.7468**	**0.0000**	**0.7468**	**0.0000**	**0.7468**	**0.0000**	**0.7468**	**0.0000**
German	0.7492	0.0065	0.7568	0.0075	0.7535	0.0067	0.7538	0.0075	**0.7575**	0.0064	0.7574	0.0065
Glass	0.6945	0.0075	**0.6955**	**0.0000**	**0.6955**	**0.0000**	**0.6955**	**0.0000**	**0.6955**	**0.0000**	**0.6955**	**0.0000**
Heart-StatLog	0.8589	0.0018	**0.8593**	**0.0000**	0.8593	0.0036	**0.8593**	**0.0000**	**0.8593**	**0.0000**	**0.8593**	**0.0000**
Hepatitis	0.7287	0.0131	0.7333	**0.0067**	**0.7353**	0.0079	0.7313	**0.0067**	0.7350	**0.0067**	0.7307	0.0080
Hillvalley	0.6244	**0.0099**	0.6169	0.0148	0.6216	0.0112	0.6279	0.0134	0.6267	0.0122	**0.6298**	0.0126
Ionosphere	0.9158	0.0100	0.9108	0.0108	0.9119	0.0102	0.9164	0.0120	**0.9181**	0.0128	0.9172	**0.0095**
Krvskp	0.9735	0.0048	**0.9801**	0.0046	0.9776	0.0050	0.9796	0.0049	0.9799	0.0049	**0.9801**	0.0050
Lung	0.9275	**0.0164**	0.9250	0.0218	0.9350	0.0217	0.9150	0.0178	0.9338	0.0228	**0.9413**	0.0213
Lung-Cancer	**0.9750**	**0.0000**	**0.9750**	**0.0000**	**0.9750**	**0.0000**	**0.9750**	**0.0000**	**0.9750**	**0.0000**	**0.9750**	**0.0000**
Lymphography	0.4333	0.0138	**0.4367**	0.0145	0.4327	0.0114	0.4327	0.0138	0.4353	**0.0112**	0.4333	**0.0112**
Movementlibras	0.8264	**0.0071**	0.8314	0.0079	0.8322	0.0073	0.8281	0.0083	**0.8347**	0.0080	0.8331	0.0091
semeion	0.9808	0.0050	**0.9821**	0.0040	0.9799	0.0026	0.9796	0.0025	0.9806	0.0034	0.9813	0.0026
Sonar	0.9167	0.0133	0.9152	0.0159	0.9238	0.0144	**0.9248**	0.0170	0.9186	**0.0120**	0.9210	0.0135
Spect	**0.7396**	0.0119	0.7404	0.0104	0.7374	0.0127	0.7385	0.0133	0.7348	**0.0091**	0.7352	0.0099
tic-tac-toe	0.8408	0.0283	0.8465	0.0204	**0.8521**	**0.0000**	**0.8521**	**0.0000**	**0.8521**	**0.0000**	**0.8521**	**0.0000**
Vehicule	0.7309	0.0094	0.7307	0.0086	0.7293	0.0105	0.7306	0.0097	0.7314	0.0104	**0.7326**	0.0054
Vowel	0.9486	0.0094	**0.9495**	**0.0000**	**0.9495**	**0.0000**	**0.9495**	**0.0000**	**0.9495**	**0.0000**	**0.9495**	**0.0000**
WaveformEW	0.8376	0.0055	0.8363	**0.0030**	0.8394	0.0043	0.8359	0.0054	0.8398	0.0045	**0.8417**	0.0060
WDBC	0.9602	0.0040	**0.9614**	0.0025	0.9612	0.0018	**0.9614**	**0.0000**	**0.9614**	**0.0000**	**0.9614**	**0.0000**
Wine	0.9550	**0.0079**	0.9544	0.0090	0.9544	0.0088	0.9550	0.0086	0.9550	0.0081	**0.9556**	0.0081
Zoo	0.9300	0.0118	0.9391	0.0101	0.9382	0.0114	0.9400	**0.0080**	0.9373	0.0100	**0.9409**	0.0112

factors to BDA, the proposed IBDA obtains the best average fitness function values on 18 datasets, which means that the introduced improved factors are effective. Note that the effectiveness of different improved factors is further verified and discussed in Section V-C. Overall, the proposed IBDA has better performance than other comparison algorithms for solving the formulated feature selection problem.

Figs. 4 and 5 show the convergence rates of different algorithms during the optimization processes. Note that these figures are separated into two part due to the space limited, and each curve is selected from the 15th test. It can be seen from these figures that the proposed IBDA expose best curves on 19 datasets, which performs the best convergence ability among all the comparison methods.

The average CPU time and Stds obtained by different algorithms for each dataset are presented in Table 5. As can be seen, although IBDA consumes more CPU time than BDA, the CPU time of IBDA is not significantly different from other algorithms, which illustrates that the overhead of IBDA is similar to the comparison algorithms. Moreover, IBDA obtains the best Std of CPU time in 16 datasets, which means that it has better computing stability in terms of CPU time.

Table 6 presents the feature selection accuracies obtained by different algorithms. As can be seen, the proposed IBDA obtains the best average accuracy results on 19 datasets. Therefore, IBDA has the best performance in terms of feature selection accuracy for these selected datasets. Moreover, Tables 7 shows the numbers of the selected features of the datasets obtained by different algorithms. It can be seen from

the table that IBDA selects more features than BDA. However, the Std of IBDA is better than BDA, which indicates it is more stable for selecting features. Note that the accuracy and number of selected features are trade-offs, which means it may be very difficult to achieve the best results in both of these two objectives for each dataset. Thus, by comprehensive considerations of the results above, we may say that the proposed IBDA employs fewer features to achieve better performance in terms of accuracy and stability. The reason may be that the introduced improved factors can enhance exploitation capabilities, thereby improving the performance of the algorithm.

C. EFFECTIVENESS OF THE PROPOSED IMPROVED FACTORS

In this section, we conduct test cases to verify the effectiveness of proposed EPD_LRS mechanism and AC operator. In the tests, we use BDA with different EPD mechanisms to solve the formulated feature selection problem for each selected dataset, respectively, and the details of these mechanisms are shown in Table 8. Similar to the previous results, the performance indexes in terms of the fitness function value, convergence rate, CPU time, accuracy and number of selection features are presented.

Table 9 shows the numerical statistical results of the average fitness function values and Stds of different approaches for each dataset, and the convergence rates during the optimization processes are shown in Figs. 6 and 7, respectively. As can be seen, the BDA_EPD_LRS mechanism achieves

TABLE 12. Number of selected features obtained by different improved factors.

	BDA_EPD		BDA_EPD_CM		BDA_EPD_RWS		BDA_EPD_Tour		BDA_EPD_LRS		BDA_EPD_LRS_AC	
	atts	std	atts	std	atts	std	atts	std	atts	std	atts	std
Arrhythmia	**75.60**	17.90	98.70	**10.68**	98.20	14.87	104.10	16.18	95.70	14.44	98.30	15.55
Breastcancer	**5.80**	0.37	6.00	**0.00**	6.00	**0.00**	6.00	**0.00**	6.00	**0.00**	6.00	**0.00**
BreastEW	3.90	1.32	2.10	0.48	**2.00**	**0.00**	**2.00**	**0.00**	**2.00**	**0.00**	**2.00**	**0.00**
Congress	6.40	1.04	6.50	0.83	**5.80**	0.83	6.20	**0.77**	6.40	0.98	6.20	0.80
Dermatology	16.50	2.11	15.50	**1.49**	15.90	2.05	**15.00**	1.67	16.10	1.75	15.90	1.49
Diabets	**4.00**	**0.00**	**4.00**	**0.00**	**4.00**	**0.00**	**4.00**	**0.00**	**4.00**	**0.00**	**4.00**	**0.00**
German	11.00	**1.37**	10.80	1.86	12.20	2.23	11.80	1.56	11.70	1.89	**10.60**	1.55
Glass	**5.00**	**0.00**	**5.00**	**0.00**	**5.00**	**0.00**	**5.00**	**0.00**	**5.00**	**0.00**	**5.00**	**0.00**
Heart-StatLog	3.30	0.45	**3.00**	**0.00**	3.10	0.25	**3.00**	**0.00**	**3.00**	**0.00**	**3.00**	**0.00**
Hepatitis	6.20	1.52	**5.50**	**0.82**	6.00	1.01	6.60	0.94	5.70	**0.82**	5.90	1.21
Hillvalley	**33.00**	7.87	43.80	8.54	42.40	**7.78**	34.80	9.16	39.50	8.24	37.00	8.12
Ionosphere	**6.00**	1.82	6.50	1.61	8.40	**1.46**	6.90	1.54	6.40	1.58	6.80	1.59
Krvskp	17.70	2.85	18.90	3.13	**16.00**	2.50	16.40	3.04	19.30	2.64	18.60	**2.46**
Lung	**94.90**	24.99	106.50	14.16	117.20	17.00	104.30	14.82	115.30	**12.19**	122.00	18.96
Lung-Cancer	8.30	3.55	6.60	**2.29**	6.70	2.68	5.70	3.15	**5.50**	2.52	6.50	3.58
Lymphography	7.50	**1.20**	7.20	1.31	**6.40**	1.83	7.60	1.23	7.10	1.91	7.20	1.78
Movementlibras	**32.30**	5.11	37.70	5.61	36.20	**3.43**	39.60	5.29	36.70	4.80	34.10	4.11
semeion	**97.00**	13.93	118.70	11.64	117.00	10.44	124.20	13.07	124.00	**9.55**	121.40	13.42
Sonar	22.60	3.74	19.90	3.62	21.70	**2.51**	19.60	2.87	**18.40**	3.12	19.30	3.44
Spect	**9.70**	2.46	10.80	**1.67**	10.80	1.68	10.20	1.70	10.70	1.74	11.70	1.80
tic-tac-toe	**8.60**	1.89	8.80	1.36	9.00	**0.00**	9.00	**0.00**	9.00	**0.00**	9.00	**0.00**
Vehicule	9.70	1.62	9.80	1.73	11.00	1.85	**9.40**	1.78	10.40	1.88	9.50	**1.38**
Vowel	**8.90**	0.06	9.00	**0.00**	9.00	**0.00**	9.00	**0.00**	9.00	**0.00**	9.00	**0.00**
WaveformEW	**19.00**	**1.84**	25.33	2.41	23.33	2.28	22.00	2.84	22.67	3.11	23.33	2.40
WDBC	3.40	0.98	2.10	0.48	2.20	0.98	**2.00**	**0.00**	**2.00**	**0.00**	**2.00**	**0.00**
Wine	5.60	1.12	5.30	**1.06**	5.30	1.13	5.50	1.14	**5.10**	1.14	5.40	1.12
Zoo	8.40	0.85	**7.10**	0.99	7.90	1.16	7.70	**0.43**	7.80	0.60	7.80	1.16

the best average fitness function value results in all EPD mechanisms, which means that EPD_LRS mechanism is more compatible with DA than other EPD mechanisms. Moreover, BDA_EPD_LRS_AC achieves the best average fitness function results on 15 datasets, which is a significant improvement over BDA_EPD_LRS. The reason may be that the AC operator can facilitate the transiting from exploration to exploitation of the search space. For references, Table 10, Table 11 and Table 12 present the summary statistical results of the CPU time, feature selection accuracies and selected number of features, respectively. It can be observed from these results that the introduced improved factors are effective for enhancing the performance of conventional DA for feature selection.

D. COMPARISON WITH OTHER FEATURE SELECTION SCHEMES

In this section, the classification of IBDA is compared to five well-known filter-based methods, which are correlation-based feature selection (CFS), fast correlation-based filter (FCBF), fisher score (F-score), information gain (IG) and wavelet power spectrum (Spectrum). Specifically, IG, Spectrum and F-Score belong to the univariate strategy, which do not reflect the dependencies of the features in the assessment measure. Moreover, CFS and FCBF belong to the multivariate strategy, which can employ the dependencies of the features. Note that these results are from [63] and the comparison results with IBDA on same datasets are shown in Table 13. It can be observed from table that IBDA can

TABLE 13. Classification accuracies achieved by different works.

	IBDA	CFS	FCBF	F-Score	IG	Spectrum
Breastcancer	0.979	0.957	**0.986**	0.979	0.957	0.957
BreastEW	**0.961**	0.825	0.798	0.930	0.930	0.772
Congress	**0.978**	0.793	0.793	0.908	0.828	0.828
Ionosphere	**0.917**	0.857	0.857	0.729	0.800	0.829
Krvskp	**0.980**	0.768	0.934	0.959	0.934	0.377
semeion	**0.981**	0.875	0.875	0.875	0.868	0.875
Sonar	**0.921**	0.310	0.214	0.048	0.191	0.048
Spect	0.735	0.736	0.774	**0.793**	**0.793**	0.736
Tic-tac-toe	**0.852**	0.000	0.000	0.010	0.010	0.167
WaveformEW	**0.842**	0.620	0.710	0.662	0.662	0.292
Wine	**0.956**	0.778	0.889	0.889	0.889	0.889
Zoo	**0.941**	0.800	0.900	0.650	0.850	0.600

outperform other algorithms on most datasets. The reason may be that the wrapper-based feature selection method can provide superior accuracy in comparison with the filter-based versions since they can utilize both labels and correlation of features.

E. LIMITATIONS OF IBDA

In this section, the limitations of the proposed IBDA is analyzed. Although the proposed IBDA outperforms some comparison algorithms according to the simulation results, it still has some limitations. First, the proposed IBDA as well as conventional DA introduces too many main parameters compared to other algorithms, which may make it difficult to tune the parameters. Second, IBDA may take more CPU time than conventional DA because the introduced improved

factors need extra computing time. For instance, the proposed EPD_LRS operator needs to relocate the worst populations of the swarm, thus the algorithm should take more CPU time. Moreover, the proposed AC operator also leads the algorithm to spend more CPU time for calculation in each loop. Finally, the Stds of fitness function values obtained by IBDA is higher than some comparison algorithms, as shown in Tables 5. This means that IBDA may lack of stability in some scenarios, and the reason may be that the mutation mechanism of the proposed EPD_LRS increases the uncertainty of the algorithm.

VI. CONCLUSION

In this paper, we investigate the feature selection problem for enhancing the classification performance in machine learning. First, a feature selection optimization problem is formulated to jointly improve the classification accuracy and reduce the number of selected features. Then, an IBDA is proposed to solve the formulated problem. IBDA introduces the EPD_LRS mechanism, AC operator and v-shape binary scheme to improve the performance of conventional DA and make it suitable for the formulated feature selection problem. By using these improved factors, the exploitation and exploration abilities of the algorithm can be balanced while the population diversity can be kept. Experiments are conducted to test the effectiveness of the proposed IBDA and the results demonstrate that it has the overall best performance on 27 well-known scientific datasets compared with BPSO, BBA, BDA, BGWO and IBPSO. Moreover, the effectiveness of the introduced improved factors are evaluated and the results verify that they are useful to enhance the performance of conventional DA for feature selection. In the future, we intend to propose more EPD strategies and combine them with other swarm intelligence algorithms to solve more optimization problems.

REFERENCES

[1] F. Cichos, K. Gustavsson, B. Mehlig, and G. Volpe, "Machine learning for active matter," *Nature Mach. Intell.*, vol. 2, no. 2, pp. 94–103, Feb. 2020.

[2] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, "Machine learning for molecular and materials science," *Nature*, vol. 559, pp. 547–555, Jul. 2018.

[3] M. Alber, A. B. Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. Karniadakis, W. W. Lytton, P. Perdikaris, L. Petzold, and E. Kuhl, "Integrating machine learning and multiscale modeling—Perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences," *npj Digit. Med.*, vol. 2, no. 1, p. 115, 2019.

[4] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*. New York, NY, USA: Springer, 1998.

[5] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 856–863.

[6] A. Bommert, X. Sun, B. Bischl, J. Rahnenführer, and M. Lang, "Benchmark for filter methods for feature selection in high-dimensional classification data," *Comput. Statist. Data Anal.*, vol. 143, Mar. 2020, Art. no. 106839.

[7] J. González, J. Ortega, M. Damas, P. Martín-Smith, and J. Q. Gan, "A new multi-objective wrapper method for feature selection–Accuracy and stability analysis for BCI," *Neurocomputing*, vol. 333, pp. 407–418, Mar. 2019.

[8] E. Ali ković and A. Subasi, "Breast cancer diagnosis using GA feature selection and rotation forest," *Neural Comput. Appl.*, vol. 28, no. 4, pp. 753–763, Apr. 2017.

[9] R. K. Huda and H. Banka, "Efficient feature selection and classification algorithm based on PSO and rough sets," *Neural Comput. Appl.*, vol. 31, no. 8, pp. 4287–4303, Aug. 2019.

[10] S. Kashef and H. Nezamabadi-pour, "An advanced ACO algorithm for feature subset selection," *Neurocomputing*, vol. 147, pp. 271–279, Jan. 2015.

[11] R. Y. M. Nakamura, L. A. M. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa, and X.-S. Yang, "BBA: A binary bat algorithm for feature selection," in *Proc. 25th SIBGRAPI Conf. Graph., Patterns Images*, Aug. 2012, pp. 291–297.

[12] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016.

[13] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, May 2016.

[14] S. R. K. S. and S. Murugan, "Memory based hybrid dragonfly algorithm for numerical optimization problems," *Expert Syst. Appl.*, vol. 83, pp. 63–78, Oct. 2017.

[15] K. Tang, X. Yuan, T. Sun, J. Yang, and S. Gao, "An improved scheme for minimum cross entropy threshold selection based on genetic algorithm," *Knowl.-Based Syst.*, vol. 24, no. 8, pp. 1131–1138, Dec. 2011.

[16] X. Guan, S. Kuang, X. Lu, and J. Yan, "Lyapunov control of high-dimensional closed quantum systems based on particle swarm optimization," *IEEE Access*, vol. 8, pp. 49765–49774, 2020.

[17] J. Luo and Z. Liu, "Novel grey wolf optimization based on modified differential evolution for numerical function optimization," *Int. J. Speech Technol.*, vol. 50, no. 2, pp. 468–486, Feb. 2020.

[18] D. Yousri, M. A. Elaziz, and S. Mirjalili, "Fractional-order calculus-based flower pollination algorithm with local search for global optimization and image segmentation," *Knowl.-Based Syst.*, vol. 197, Jun. 2020, Art. no. 105889.

[19] R. Salem, M. Abdul Salam, H. Abdelkader, and A. Awad Mohamed, "An artificial bee colony algorithm for data replication optimization in cloud environments," *IEEE Access*, vol. 8, pp. 51841–51852, 2020.

[20] D. Wang, X. Qian, X. Ban, B. Ma, Y. Ma, and Z. Lv, "Enhanced bacterial foraging optimization based on progressive exploitation toward local optimum and adaptive raid," *IEEE Access*, vol. 7, pp. 95725–95738, 2019.

[21] A. M. Eltamaly, M. S. Al-Saud, and A. G. Abokhalil, "A novel bat algorithm strategy for maximum power point tracker of photovoltaic energy systems under dynamic partial shading," *IEEE Access*, vol. 8, pp. 10048–10060, 2020.

[22] G. Sun, X. Zhao, S. Liang, Y. Liu, X. Zhou, and Y. Zhang, "A modified chicken swarm optimization algorithm for synthesizing linear, circular and random antenna arrays," in *Proc. IEEE 90th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2019, pp. 1–7.

[23] J. Zhao, W. Chen, J. Ye, H. Wang, H. Sun, and I. Lee, "Firefly algorithm based on level-based attracting and variable step size," *IEEE Access*, vol. 8, pp. 58700–58716, 2020.

[24] Q. Han, X. Yang, H. Song, S. Sui, H. Zhang, and Z. Yang, "Whale optimization algorithm for ship path optimization in large-scale complex marine environment," *IEEE Access*, vol. 8, pp. 57168–57179, 2020.

[25] X. Yue, H. Zhang, and H. Yu, "A hybrid grasshopper optimization algorithm with invasive weed for global optimization," *IEEE Access*, vol. 8, pp. 5928–5960, 2020.

[26] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.

[27] G. Dhiman and V. Kumar, "Emperor penguin optimizer: A bio-inspired algorithm for engineering problems," *Knowl.-Based Syst.*, vol. 159, pp. 20–50, Nov. 2018.

[28] G. Dhiman and V. Kumar, "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems," *Knowl.-Based Syst.*, vol. 165, pp. 169–196, Feb. 2019.

[29] Anita and A. Yadav, "AEFA: Artificial electric field algorithm for global optimization," *Swarm Evol. Comput.*, vol. 48, pp. 93–108, Aug. 2019.

[30] L. Zhihui, C. Qian, Z. Yonghua, T. Pengfei, and Z. Rui, "Krill herd algorithm for signal optimization of cooperative control with traffic supply and demand," *IEEE Access*, vol. 7, pp. 10776–10786, 2019.

[31] G. Dhiman and V. Kumar, "Spotted hyena optimizer: A novel bio-inspired based Metaheuristic technique for engineering applications," *Adv. Eng. Softw.*, vol. 114, pp. 48–70, Dec. 2017.

[32] H. Singh Gill, B. Singh Khehra, A. Singh, and L. Kaur, "Teaching-learning-based optimization algorithm to minimize cross entropy for selecting multilevel threshold values," *Egyptian Informat. J.*, vol. 20, no. 1, pp. 11–25, Mar. 2019.

[33] A. García-Dominguez, C. E. Galván-Tejada, L. A. Zanella-Calzada, H. Gamboa-Rosales, J. I. Galván-Tejada, J. M. Celaya-Padilla, H. Luna-García, and R. Magallanes-Quintanar, "Feature selection using genetic algorithms for the generation of a recognition and classification of children activities model using environmental sound," *Mobile Inf. Syst.*, vol. 2020, Feb. 2020, Art. no. 8617430.

[34] X. Liu and L. Shang, "A fast wrapper feature subset selection method based on binary particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 3347–3353.

[35] F. Liu, X. Yan, and Y. Lu, "Feature selection for image steganalysis using binary bat algorithm," *IEEE Access*, vol. 8, pp. 4244–4249, 2020.

[36] K. Devanathan, N. Ganapathy, and R. Swaminathan, "Binary grey wolf optimizer based feature selection for nucleolar and centromere staining pattern classification in indirect immunofluorescence images," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2019, pp. 7040–7043.

[37] D. Rodrigues, L. A. M. Pereira, T. N. S. Almeida, J. P. Papa, A. N. Souza, C. C. O. Ramos, and X.-S. Yang, "BCS: A binary cuckoo search algorithm for feature selection," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2013, pp. 465–468.

[38] H. M. Zawbaa, E. Emary, B. Parv, and M. Sharawi, "Feature selection approach based on moth-flame optimization algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 4612–4617.

[39] E. Zorarpaci and S. A. Özel, "A hybrid approach of differential evolution and artificial bee colony for feature selection," *Expert Syst. Appl.*, vol. 62, pp. 91–103, Nov. 2016.

[40] P. Shunmugapriya and S. Kanmani, "A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid)," *Swarm Evol. Comput.*, vol. 36, pp. 27–36, Oct. 2017.

[41] Q. Al-Tashi, S. J. Abdul Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Binary optimization using hybrid grey wolf optimization for feature selection," *IEEE Access*, vol. 7, pp. 39496–39508, 2019.

[42] K. K. Bharti and K. K. Bharti, "A novel hybrid BPSO—A approach for feature selection," *Natural Comput.*, pp. 1–23, Oct. 2019, doi: 10.1007/s11047-019-09769-z.

[43] M. Mafarja, A. Qasem, A. A. Heidari, I. Aljarah, H. Faris, and S. Mirjalili, "Efficient hybrid nature-inspired binary optimizers for feature selection," *Cognit. Comput.*, vol. 12, no. 1, pp. 150–175, Jan. 2020.

[44] Y. Zhang, X.-F. Song, and D.-W. Gong, "A return-cost-based binary firefly algorithm for feature selection," *Inf. Sci.*, vols. 418–419, pp. 561–574, Dec. 2017.

[45] H. Faris, M. M. Mafarja, A. A. Heidari, I. Aljarah, A. M. Al-Zoubi, S. Mirjalili, and H. Fujita, "An efficient binary salp swarm algorithm with crossover scheme for feature selection problems," *Knowl.-Based Syst.*, vol. 154, pp. 43–67, Aug. 2018.

[46] L. Zhu, S. He, L. Wang, W. Zeng, and J. Yang, "Feature selection using an improved gravitational search algorithm," *IEEE Access*, vol. 7, pp. 114440–114448, 2019.

[47] I. Tumar, Y. Hassouneh, H. Turabieh, and T. Thaher, "Enhanced binary moth flame optimization as a feature selection algorithm to predict software fault prediction," *IEEE Access*, vol. 8, pp. 8041–8055, 2020.

[48] M. Tubishat, N. Idris, L. Shuib, M. A. M. Abushariah, and S. Mirjalili, "Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection," *Expert Syst. Appl.*, vol. 145, May 2020, Art. no. 113122.

[49] S. Sankhwar, D. Gupta, K. C. Ramya, S. Sheeba Rani, K. Shankar, and S. K. Lakshmanaprabu, "Improved grey wolf optimization-based feature subset selection with fuzzy neural classifier for financial crisis prediction," *Soft Comput.*, vol. 24, no. 1, pp. 101–110, Jan. 2020.

[50] H. Ghimatgar, K. Kazemi, M. S. Helfroush, and A. Aarabi, "An improved feature selection algorithm based on graph clustering and ant colony optimization," *Knowl.-Based Syst.*, vol. 159, pp. 270–285, Nov. 2018.

[51] M. Tubishat, M. A. M. Abushariah, N. Idris, and I. Aljarah, "Improved whale optimization algorithm for feature selection in arabic sentiment analysis," *Int. J. Speech Technol.*, vol. 49, no. 5, pp. 1688–1707, May 2019.

[52] H. Dong, J. Sun, T. Li, and L. Li, "An improved niching binary particle swarm optimization for feature selection," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2018, pp. 3571–3577.

[53] B. Ji, X. Lu, G. Sun, W. Zhang, J. Li, and Y. Xiao, "Bio-inspired feature selection: An improved binary particle swarm optimization approach," *IEEE Access*, vol. 8, pp. 85989–86002, 2020.

[54] A. Adeli and A. Broumandnia, "Image steganalysis using improved particle swarm optimization based feature selection," *Appl. Intell.*, vol. 48, no. 6, pp. 1609–1622, 2018.

[55] M. M. Mafarja, D. Eleyan, I. Jaber, A. Hammouri, and S. Mirjalili, "Binary dragonfly algorithm for feature selection," in *Proc. Int. Conf. New Trends Comput. Sci. (ICTCS)*, Oct. 2017, pp. 12–17.

[56] E. Elhariri, N. El-Bendary, and A. E. Hassanien, "Bio-inspired optimization for feature set dimensionality reduction," in *Proc. 3rd Int. Conf. Adv. Comput. Tools for Eng. Appl. (ACTEA)*, Jul. 2016, pp. 184–189.

[57] M. A. Tawhid and K. B. Dsouza, "Hybrid binary dragonfly enhanced particle swarm optimization algorithm for solving feature selection problems," *Math. Found. Comput.*, vol. 1, no. 2, pp. 181–200, 2018.

[58] M. Hariharan, R. Sindhu, V. Vijean, H. Yazid, T. Nadarajaw, S. Yaacob, and K. Polat, "Improved binary dragonfly optimization algorithm and wavelet packet based non-linear features for infant cry classification," *Comput. Methods Programs Biomed.*, vol. 155, pp. 39–51, Mar. 2018.

[59] G. I. Sayed, A. Tharwat, and A. E. Hassanien, "Chaotic dragonfly algorithm: An improved metaheuristic algorithm for feature selection," *Appl. Intell.*, vol. 49, no. 1, pp. 188–205, 2019.

[60] B. Xue, M. Zhang, and W. N. Browne, "A comprehensive comparison on evolutionary feature selection approaches to classification," *Int. J. Comput. Intell. Appl.*, vol. 14, no. 02, Jun. 2015, Art. no. 1550008.

[61] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statistician*, vol. 46, no. 3, pp. 175–185, Aug. 1992.

[62] P. Bak, C. Tang, and K. Wiesenfeld, "Self-organized criticality: An explanation of the $1/f$ noise," *Phys. Rev. Lett.*, vol. 59, no. 4, pp. 381–384, 1987.

[63] M. Mafarja, I. Aljarah, A. A. Heidari, A. I. Hammouri, H. Faris, A. M. Al-Zoubi, and S. Mirjalili, "Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems," *Knowl.-Based Syst.*, vol. 145, pp. 25–45, Apr. 2018.

[64] Y. Hou, J. Li, H. Yu, and Z. Li, "BIFFOA: A novel binary improved fruit fly algorithm for feature selection," *IEEE Access*, vol. 7, pp. 81177–81194, 2019.

[65] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation Metaheuristic algorithms," *Comput. Ind. Eng.*, vol. 137, Nov. 2019, Art. no. 106040.

[66] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Inf. Sci.*, vol. 237, pp. 82–117, Jul. 2013.

[67] B. Tran, B. Xue, and M. Zhang, "Variable-length particle swarm optimization for feature selection on high-dimensional classification," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 473–487, Jun. 2019.

[68] C. Huang, Y. Li, and X. Yao, "A survey of automatic parameter tuning methods for metaheuristics," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 201–216, Apr. 2020.

[69] S.-W. Lin, K.-C. Ying, S.-C. Chen, and Z.-J. Lee, "Particle swarm optimization for parameter determination and feature selection of support vector machines," *Expert Syst. Appl.*, vol. 35, no. 4, pp. 1817–1824, Nov. 2008.

[70] M. Mafarja, I. Aljarah, A. A. Heidari, H. Faris, P. Fournier-Viger, X. Li, and S. Mirjalili, "Binary dragonfly optimization for feature selection using time-varying transfer functions," *Knowl.-Based Syst.*, vol. 161, pp. 185–204, Dec. 2018.

JIAHUI LI received the B.S. degree in software engineering from Jilin University, Changchun, China, in 2018, where he is currently pursuing the master's degree in computer science and technology. His current research interests include edge computing and feature engineering and optimization.

HUI KANG received the M.E. and Ph.D. degrees from Jilin University, in 1996 and 2007, respectively. She is currently an Associate Professor with the College of Computer Science and Technology, Jilin University. Her research interests include information integration and distributed computing.

WENQI LI received the B.S. degree in software engineering from Jilin University, in 2019, where he is currently pursuing the master's degree in computer science and technology. His research interests include optimization, swarm intelligence, and feature selection.

GENG SUN (Member, IEEE) received the B.S. degree in communication engineering from Dalian Polytechnic University, in 2007, and the Ph.D. degree in computer science and technology from Jilin University, in 2018. He was a Visiting Researcher with the School of Electrical and Computer Engineering, Georgia Institute of Technology, USA. He currently holds a postdoctoral position with Jilin University. His research interests include wireless sensor networks, antenna arrays, collaborative beamforming, and optimization.

WEI ZHANG received the B.S. degree in clinical medicine and the master's degree in hepatobiliary and pancreatic surgery from Jilin University, in 2000 and 2007, respectively. He was a Visiting Researcher with Stanford University, USA, in 2019. He is currently a Chief Surgeon with Jilin University. His research interests include liver cancer and laparoscopic therapy.

TIE FENG received the M.S. and Ph.D. degrees from Jilin University, in 1994 and 2007, respectively. He is currently an Associate Professor with the College of Computer Science and Technology, Jilin University. His research interests include software maintenance and evolution, reverse engineering, software refactoring, and automated software engineering.

BAI JI received the B.S. degree in clinical medicine and the Ph.D. degree in hepatobiliary and pancreatic surgery from Jilin University, in 2003 and 2013, respectively. She was a Visiting Researcher with the Mount Sinai Hospital, New York, NY, USA, in 2013. She was a Visiting Researcher with the University of Minnesota, in 2016. She is currently a Professor with Jilin University. Her research interests include machine learning in liver cancer, chemotherapy resistance, and targeted therapy.

• • •