Bio-Inspired Feature Selection: An Improved Binary Particle Swarm Optimization Approach

BAI JI¹, XIAOZHENG LU¹, GENG SUN²,³, (Member, IEEE), WEI ZHANG¹, JIAHUI LI², AND YINZHE XIAO²

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

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

This work was supported in part by the National Natural Science Foundation of China under Grant 61872158, Grant 61806083, and Grant 81802805, in part by the Postdoctoral Innovative Talent Support Program of China under Grant BX2018128, in part by the China Postdoctoral Science Foundation under Grant 2018M640283, in part by the Science and Technology Development Plan Project of Jilin Province under Grant 20190701019GH, and in part by the China Guanghua Science and Technology Foundation of the First Hospital of Jilin University under Grant JDYYGH2019031.

ABSTRACT Feature selection is an effective approach to reduce the number of features of data, which enhances the performance of classification in machine learning. In this paper, we formulate a joint feature selection problem to reduce the number of the selected features while enhancing the accuracy. An improved binary particle swarm optimization (IBPSO) algorithm is proposed to solve the formulated problem. IBPSO introduces a local search factor based on Lévy flight, a global search factor based on weighting inertia coefficient, a population diversity improvement factor based on mutation mechanism and a binary mechanism to improve the performance of conventional PSO and to make it suitable for the binary feature selection problems. Experiments based on 16 classical datasets are selected to test the effectiveness of the proposed IBPSO algorithm, and the results demonstrate that IBPSO has better performance than some other comparison algorithms.

INDEX TERMS Feature selection, classification, bio-inspired computing, particle swarm optimization.

I. INTRODUCTION

Machine learning has been widely applied in many practical applications such as data mining, text processing, pattern recognition and medical image analysis, and these fields often rely on the datasets with a large amount of data [1]. However, part of the features may be irrelevant or even misleading for the machine learning algorithms, which increase the computational overhead and reduce accuracy of classification especially for the high-dimensional datasets [2], [3]. Thus, it is necessary to conduct feature selections.

The main principle of feature selection is to find an optimal subset of features which is discriminating from the full dataset, and the selected subset should remain or even enhance the classification performance of the original dataset [4]. Feature selections are useful methods because they can eliminate redundant noise from the datasets so that making the machine learning algorithms perform to execute faster and more efficient. In other words, by using feature selection, the machine learning approaches may perform better while saving costs [5].

According to the principles, feature selection methods mainly include three categories that are the filters, wrapper and embedded approaches [6]. The filter approaches use a statistical measure to assign a relevance score to each feature and rank the features according to the computed scores [7]. The wrapper methods adopt classifiers to evaluate the selected subsets obtained by selection algorithms, and use the feedback of classifiers to guide the feature selection. Thus, accuracies of the wrapper methods are better than the filter methods [8]. However, they may consume more computing resources. Moreover, the embedded methods are actually the special cases of the wrapper methods since the feature selections are regarded as a part of training phase in machine learning [9].

Generally, feature selections can be regarded as the optimization problems in which a subset of the original dataset is represented by a solution to the optimization problem, and these problems can be solved by exhaustive and heuristic search approaches [10]. However, the computation costs are
usually unacceptable by using the exhaustive search methods, especially for the dataset with high dimensions of data. Therefore, the heuristic approaches may be more reasonable methods for solving the feature selection problems. Swarm intelligence algorithms are efficient heuristic search methods for the wrapper-based feature selection problems [11]. For example, several classical swarm intelligence algorithms such as genetic algorithm (GA) [12], ant colony optimization (ACO) [13], differential evolution (DE) [14], grey wolf optimization (GWO) [15] and dragon algorithm (DA) [16].

Among these kind of algorithms, the particle swarm optimization (PSO) algorithm is demonstrated as an effective method for the optimization problems since it is powerful while easy to be implemented [17]. However, conventional PSO algorithm may have certain drawbacks such as it is lack of exploitation for some problems. Moreover, according to the no-free-lunch (NFL) theory, there will be no algorithm that is suitable for solving all the optimization problems [18]. In addition, the conventional PSO is proposed for the continuous optimization problem, which cannot be used for the feature selection problems with binary solution space. Thus, the conditions above motive us to improve the conventional PSO so that making it more suitable for the feature selection.

The main contributions of this paper are summarized as follows:

- We formulate a joint feature selection problem to simultaneously reduce the number of selection features and enhance the accuracy.
- We propose an improved binary PSO (IBPSO) algorithm to solve the formulated feature selection problem. IBPSO introduces a local search operator, a global search operator, a population diversity improvement factor and a binary mechanism to improve the performance of conventional PSO and to make it suitable for the binary feature selection problem.
- The performance of the proposed IBPSO is verified by 16 classical datasets, and several other algorithms are selected for comparisons.

A. ROADMAP

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 IBPSO algorithm. Section V shows the experiment results and Section VI presents a summary of findings and conclusions.

II. RELATED WORK

In this section, the previous works that related to feature selections based on various methods are reviewed. Several works that are not based on swarm intelligence and evolutionary algorithms are proposed for feature selections, e.g., the works in references [19], [20] and [21]. Recently, the swarm intelligence and evolutionary algorithms may be more popular approaches for feature selections [11].

Genetic algorithms (GAs) may be the first technologies that are widely adopted in feature selections [22], and many GA-based methods are proposed for these problems [23]. For example, Sayed et al. [24] propose a nested GA for feature selection in high-dimensional cancer microarray datasets. Liu et al. propose a hybrid GA with wrapper-embedded approach to select features from several datasets. In addition to GA, other swarm intelligence algorithms are also used for feature selections. Mistry et al. [25] propose a micro-GA embedded PSO feature selection method for the intelligent facial emotion recognition problems. Zhu et al. [26] propose an improved gravitational search algorithm to solve the feature selection problems, and the effectiveness of the proposed method are evaluated on several widely used datasets. Taradeh et al. [27] propose another gravitational search-based algorithm for feature selection. Emary et al. [28] use the firefly algorithm (FA) to select features from several datasets. In [29], the authors propose to use an ant colony optimization (ACO) with support vector machine (SVM) strategy for wrapper feature selection in face recognition. O’Boyle et al. [30] use ACO to select features and optimize the parameters of an SVM system, in which a weighting method is adopted for improving the performance of the algorithm. Moreover, Rodrigues et al. [31] use the cuckoo search (CS) algorithm to solve the feature selection problems.

Recently, the enhanced versions of swarm intelligence and evolutionary algorithms are more popular methods for feature selections. Dong et al. [32] propose an improved binary GA with feature granulation to select the significant features in datasets. Hou et al. [18] use a novel binary improved fruit fly optimization (FFO) algorithm to improve the performance of conventional FFO algorithm for feature selections. The authors in [33] combine the differential evolution (DE) with ACO for feature selection problems, and DE is adopted to further search for optimal feature subset based on the solutions generated by ACO. Aziz and Hassnien [34] propose a modified CS method which combines rough sets for feature selections. Zhang et al. [35] propose a new version of FFO algorithm to solve the feature selection problem. In this approach, the Gaussian mutation operator and chaotic local search method are introduced to enhance the performance of the original algorithm. Anter et al. [36] use a chaotic binary grey wolf optimization (GWO) approach as the feature selection model that attempts to reduce the number of features without loss of significant information for classification. Abdel-Basset et al. [37] propose a GWO algorithm integrated with a two-phase mutation strategy to solve the feature selection for classification problems based on the wrapper methods, and the two-phase mutation enhances the exploitation capability of the algorithm. Reference [38] proposes a brain storm optimization with a new individual clustering technology and two individual updating mechanisms for developing novel feature selection algorithms with the purpose of maximizing the classification performance. More improved algorithms for feature selections can be found in [39]–[42], [43] and [44].
For the feature selections that based on PSO approach, Zhang et al. [45] propose an improved multi-objective PSO for multi-label feature selection. In their method, two new operators that are adaptive uniform mutation and local learning strategy are introduced to enhance the performance of the algorithm. Qi et al. [46] use PSO with mutation mechanism and the SVM for feature selection in hyperspectral classification. Chhikara et al. [47] propose a feature selection approach based on an improved PSO algorithm and filter approaches to enhance the classification accuracy and reduce the computational complexity in image steganalysis. The authors in [48] utilizes a filter-based feature selection technique which uses an information theoretic-PSO approach to determine the most optimal feature combination in biomedical entity extraction. Chen et al. [49] propose a hybrid PSO with a spiral-shaped mechanism for selecting the optimal feature subset for classification via a wrapper-based approach. Ding et al. [50] add the crossover and mutation operators of GA to the competitive swarm optimization which is an extended version of PSO to the competitive swarm optimization. Thus, in this work, a novel IBPSO by balancing the exploration and exploitation is proposed for trying to deal with more feature selection problems.

### III. PROBLEM FORMULATION

The goal of feature selection in this work is to reduce the number of selected features while maximizing the classification accuracy, which can be regarded as a multi-objective optimization problem. To consider these two objectives, we design the fitness function based on the linear weighting method as follows:

$$f_{FS} = a \cdot E + b \cdot \frac{R}{C}$$  \hspace{1cm} (1)

where $E$ is the classification error rate of a certain classifier, $R$ and $C$ represent the number of selected features and the total number of features, respectively. Moreover, $a$ and $b$ are the weights which are used to balance these two objectives.

Note that the $k$-nearest neighbor (KNN) method is applied to implement wrapper method and evaluate the classification accuracy in this work. KNN is a classical and popular machine learning algorithm, which keeps all the training data for classification. The details of KNN can be learned in [52]. Moreover, it should be also noted that the classifier is not the contributions of this work, and we only adopt the classification results as the indicators.

### IV. PROPOSED IBPSO

The search space in the formulated feature selection problem is nonlinear and discrete, and there may be a large number of local minimum points. Thus, we propose an IBPSO algorithm for solving the feature selection problems and the details of this algorithm are as follows.

### A. CONVENTIONAL PSO

The conventional PSO is a bio-inspired evolutionary algorithm which is inspired by the food searching behaviors of fish or birds in nature. In this algorithm, each candidate solution of the optimization problem is represented by a particle, and each particle has two main properties that are position and velocity. For an optimization problem with $n$ dimensions, the $i$th particle moves with a certain velocity $v_i$ and the position of the particle is expressed as $x_i$. Thus, a solution and its position in PSO algorithm can be written as: $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$ and $v_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,n})$, respectively.

In PSO, the quality of a solution is evaluated by the fitness function and a particle with better fitness function value is regarded as a better solution. In each iteration, the position and velocity of each particle in the population should be updated to generate new solutions, and the solution update method is as follows [17]:

$$x_{i}^{t+1} = x_{i}^{t} + v_{i}^{t+1}$$  \hspace{1cm} (2)

$$v_{i}^{t+1} = \omega \cdot v_{i}^{t} + c_1 \cdot r_1 \cdot (P_{best}^t - x_{i}^{t}) + c_2 \cdot r_2 \cdot (G_{best} - x_{i}^{t})$$  \hspace{1cm} (3)

where $P_{best}^t$ is the personal best solution which means it is a solution with best fitness function value in $t$th iteration, $G_{best}$ is the global best solution, which mean it is the solution with the best fitness function value currently. $\omega$ is the inertia coefficient which ranges from 0 to 1, $c_1$ and $c_2$ are the accelerating coefficients. Moreover, $r_1$ and $r_2$ are random numbers generated by uniform distributions and the ranges of them are both $[0, 1]$.

Accordingly, the main steps of conventional PSO are shown in Fig. 1, and the details are further explained as follows:

- **Step 1:** Define the key parameters such as the population size $N_{pop}$ and fitness function $f(x)$, and initialize the positions and velocities of the particles randomly.
- **Step 2:** Calculate the fitness function value of each particle and sort the solutions according these values.
- **Step 3:** Update $P_{best}$ and $G_{best}$ based on the fitness function values.
- **Step 4:** Update the position and velocity of each particle by Eqs. 2 and 3.
- **Step 5:** If the stopping criteria is reached, then the algorithm finishes. Otherwise, go to Step 2 for a loop.

### B. IBPSO

It has been reported that conventional PSO algorithm has some drawbacks for certain optimization problems.
For example, it may have a big possibility to fall into local optima. Moreover, feature selection is a binary optimization problem by nature, which means that the conventional PSO algorithm can not be used for this problem directly since it is proposed for continuous problems. Thus, we propose an IBPSO algorithm for solving the formulated feature selection problem. IBPSO introduces the local search, global search and population diversity improvement factors to improve the performance of conventional PSO, and uses a binary mechanism to make the algorithm suitable for feature selection.

Note that the basic principle of IBPSO is to let the searching agents of the continuous algorithm to move around the search space continuously, and then map the obtained continuous solutions into a binary space. The pseudo code of the proposed IBPSO is shown in Algorithm 1, and the details of the introduced improved factors are as follows.

1) LOCAL SEARCH BASED ON LÉVY FLIGHT
In conventional PSO, each solution is updated by the guidance of \( G_{\text{best}} \), so that it may move toward a better position. However, if \( G_{\text{best}} \) is with a position that far away from the optimal solution, the algorithm may fall into local optima, which reduces the performance of the algorithm. Thus, to further enhance the exploitation ability of conventional PSO, a local search operator based on Lévy flight mechanism for \( G_{\text{best}} \) is proposed. Lévy flight is a random walk method which follows a heavy-tailed distribution. In this method, the short-distance and occasional long-distance searching appear alternately, such that expands the search scope and enhances the local search performance.

Accordingly, the proposed Lévy flight-based local search factor is described as follows [53]:

\[
G_{\text{best}}^{\text{new}} = G_{\text{best}} + \alpha \oplus \text{Lévy}(\lambda)
\] (4)

2) GLOBAL SEARCH BASED ON WEIGHTING INERTIA COEFFICIENT
In conventional PSO, the inertia coefficient is an important parameter that affects the searching performance of the algorithm since it is able to determine the searching scope. However, the value of inertia coefficient is fixed during the iterative process, which may be not suitable since the required searching steps of the initial and end stages are different. Thus, to overcome this shortcoming, we propose a weighting inertia coefficient to PSO and it is described as follows:

\[
\omega_{w}^{t} = (\omega_{\text{max}} - \omega_{\text{min}}) \cdot \frac{t + t_{\text{max}}}{2 \times t_{\text{max}}}
\] (6)

where \( \omega_{\text{max}} \) and \( \omega_{\text{min}} \) are the maximum and minimum values of inertia coefficient. Then, the position update method by introducing the weighting inertia coefficient is as follows:

\[
v_{i+1}^{t} = \omega_{w}^{t} \cdot v_{i}^{t} + c_{1} \cdot r_{1} \cdot (P_{\text{best}_{i}}^{t} - x_{i}^{t}) + c_{2} \cdot r_{2} \cdot (G_{\text{best}}^{t} - x_{i}^{t})
\] (7)

Algorithm 1  IBPSO

1. Define and initialize the related parameters: population size \( N_{\text{pop}} \), solution dimension \( N_{\text{dim}} \), maximum iteration \( t_{\text{max}} \) and fitness function, etc.;
2. Map the searching agents into binary spaces by using Eqs. (11) and (11);
3. Calculate the fitness function values of mapped particles and sort these particles in ascending order;
4. for \( t = 1 \) to \( t_{\text{max}} \) do
   5. for \( i = 1 \) to \( N_{\text{pop}} \) do
      6. Update each searching agent particles by using Eq. (6);
      7. Apply the local search operator to update \( G_{\text{best}} \) by using Eqs. (4);
      8. if \( G_{\text{best}}^{\text{new}} \) is better than \( G_{\text{best}} \) then
         9. \( G_{\text{best}} \) is replaced by \( G_{\text{best}}^{\text{new}} \);
      else
         10. Remain \( G_{\text{best}} \);
   end
5. end
6. Mutate solutions by using Algorithm 2;
7. Map the continuous searching agents into binary spaces by using Eqs. (11) and (11);
8. Calculate the fitness function values of mapped particles;
9. end
10. Return \( x_{\text{best}} \)
11. //\( x_{\text{best}} \) is the best solution obtained by the algorithm

where \( \alpha \) is the step factor and its value depends on the applications. Moreover, the random step value of Lévy flight is taken from the Lévy distribution, which is:

\[
\text{Lévy}(\lambda) \sim u = t^{-\lambda}(1 < \lambda < 3)
\] (5)
By using the weighting inertia coefficient, the searching step can be dynamically adjusted according to the iterations of algorithm, that is, if the algorithm is in the initial stage, it may be better to have a longer searching step since the current solutions may be far away from the optimal location and vice versa. Thus, by using the weighting inertia coefficient, the global search ability of the algorithm may be improved.

3) POPULATION DIVERSITY IMPROVEMENT BASED ON MUTATION MECHANISM

The feature selection problems are usually with high solution dimensions, which cause the algorithm has a huge population. Moreover, since the solutions in conventional PSO are updated by the guidance of the corresponding \( P_{best} \), which may cause the solutions with poor \( P_{best} \) to be over-exploited in a low-fitness solution space, so that reducing exploitation efficiency. However, if the weight of \( G_{best} \) is increased to solve this issue, then all solutions are updated by the guidance of \( G_{best} \), causing the updated solutions may be similar with each other, so that reducing the population diversity. Therefore, it may be difficult to maintain both exploitation efficiency and population diversity in conventional PSO.

To overcome this issue above, we introduce a mutation mechanism to mutate some of the solutions and the corresponding \( P_{best} \) values of them in the population, thereby enhancing the development efficiency while ensuring the population diversity. In this work, half of the solutions that with the worst fitness function values and their corresponding \( P_{best} \) are selected to be mutated by using the method as follows:

\[
P^{t+1}_{best_i} = P^{t}_{best_{i1}} + M \cdot (P^{t}_{best_{i2}} - P^{t}_{best_{i3}}) \tag{8}
\]

where \( P^{t}_{best_{i1}} \), \( P^{t}_{best_{i2}} \) and \( P^{t}_{best_{i3}} \) are the randomly selected personal best solutions in the first half of population which with better fitness function values, \( m_1 \neq m_2 \neq m_3 \), \( M \) is an adjust factor. By using this mutation operator, the solutions that with worst fitness function values can be further guided by the personal best solutions with better fitness function values, which may improve the exploitation efficiency.

Moreover, the solutions in the second half of population with worse fitness function values should be also mutated to further improve the population diversity. Thus, we propose to use a threshold \( \varphi \) to determine which dimension of solution is updated, and \( \varphi \) is defined as follows:

\[
\varphi = 0.8 - \frac{0.79}{1 + e^{-\frac{t}{t_{max}}}} \tag{9}
\]

where \( t \) is the current iteration number. In our scheme, if a randomly generated number \( rand \) is less than \( \varphi \), then the corresponding dimension of solution should be mutated to its inverse value by using Eq. (10) as follows:

\[
x^d = 1 - x^d \tag{10}
\]

where \( x^d \) represent the \( d \)th dimension of a solution. The main steps of mutation mechanism is shown in Algorithm 2.

4) BINARY MECHANISM

A binary mechanism 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 widely used Sigmoid function is adopted to IBPSO for the solution mappings, and the details of this function is as follows [54]:

\[
x_{sig} = \frac{1}{1 + e^{-x_{binary} + 1}} \tag{11}
\]

\[
x_{binary} = \begin{cases} 1, & N_{random} \leq x_{sig} \\ 0, & N_{random} > x_{sig} \end{cases} \tag{12}
\]

where \( x_{binary} \) is the converted binary solution of the feature selection problem, and \( N_{random} \) is a random number which is used as the threshold.

C. FEATURE SELECTION BASED ON IBPSO

To solve the formulated feature selection problem by using the proposed IBPSO, a solution can be regarded as a particle. Thus, the solutions can be expressed as follows:

\[
x = (B_1, B_2, B_3, \cdots, B_n) \tag{13}
\]

where \( n \) represents the number of features. Correspondingly, the population of IBPSO is expressed as follows:

\[
\text{pop} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N_{pop}} \end{bmatrix} = \begin{bmatrix} B_1^1 & B_1^2 & B_1^3 & \cdots & B_1^n \\ B_2^1 & B_2^2 & B_2^3 & \cdots & B_2^n \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ B_{N_{pop}}^1 & B_{N_{pop}}^2 & B_{N_{pop}}^3 & \cdots & B_{N_{pop}}^n \end{bmatrix} \tag{14}
\]

D. COMPUTATIONAL COMPLEXITY

The complexity of the proposed IBPSO is analyzed in this section. We also suppose that the maximum number of iteration and population size are \( t_{max} \) and \( N_{pop} \), respectively, then
the complexity of conventional PSO algorithm is $O(t_{\text{max}} \cdot N_{\text{pop}})$ because there is only one inner loop of the algorithm. Since the structure of the proposed IBPSO is similar with PSO, the complexity of IBPSO is also $O(t_{\text{max}} \cdot N_{\text{pop}})$. However, IBPSO may consume more computing time than conventional binary PSO (BPSO) for solving a certain feature selection problem even if they have the same computational complexity. The reason may be that the introduced improved factors lead to additional computing time, and this will be further evaluated and discussed in the following section.

V. RESULTS AND ANALYSIS

In this section, we conduct tests to evaluate the performance of the proposed IBPSO algorithm for feature selections. Moreover, several other algorithms are selected for comparisons.

A. BENCHMARK DATASETS AND EXPERIMENT SETUPS

The benchmark datasets used in the evaluations and parameter setups of different algorithms are introduced.

1) BENCHMARK DATASETS

In this work, we select 16 datasets from the widely used UC Irvine Machine Learning Repository [55], and the main information of these selected datasets are shown in Table 1.

| Number | Dataset       | Number of feature | Number of instance |
|--------|---------------|-------------------|-------------------|
| 1      | Breastcancer  | 9                 | 699               |
| 2      | BreastEW      | 30                | 596               |
| 3      | Exactly       | 13                | 1000              |
| 4      | Exactly2      | 13                | 1000              |
| 5      | HeartEW       | 13                | 270               |
| 6      | Lymphography  | 18                | 148               |
| 7      | SonarEW       | 60                | 208               |
| 8      | SpectrEW      | 22                | 267               |
| 9      | CongressEW    | 16                | 435               |
| 10     | IonosphereEW  | 34                | 351               |
| 11     | KruskalEW     | 36                | 3196              |
| 12     | Tic-tac-toe   | 9                 | 958               |
| 13     | Vote          | 16                | 300               |
| 14     | WavelengthEW  | 40                | 5000              |
| 15     | Zoo           | 16                | 101               |
| 16     | Lungcancer    | 56                | 32                |

2) PARAMETER TUNING

As we known, according to the NFL theory, it is difficult for a metaheuristics algorithm to perform excellent performance on all the optimization problems, especially by using the same parameter setups. Thus, it is better to tune the key parameters for each optimization problem separately so that achieving the best performance. However, in this work, the number of features (solution dimensions) of these datasets are quite different, which means that the formulated feature selection optimization problem for each dataset may be regarded as independent optimization problems. Thus, it is better to tune the key parameters of IBPSO for each dataset. However, this will be huge works since there are 16 datasets that need to be tuned in our work. Therefore, refer to [56], we selected BreastEW dataset to tune the key parameters of the proposed IBPSO since this dataset has the median size compared to other datasets.

In the tuning test, the accelerating coefficients $c_1$ and $c_2$, which are the key parameters of PSO as well as IBPSO, are jointly tuned. Specifically, we used the proposed IBPSO with different combinations of $c_1$ and $c_2$ to select the features on BreastEW dataset, and the ranges of these two parameters are both 1 to 2 with the step size of 0.25, resulting in 16 different combinations of these parameters. Moreover, each combination is independently run for 30 time to avoid random bias, and the average results are presented. The parameter tuning results are shown in Table 2. According to the results, when $c_1 = 1.75$ and $c_2 = 2.0$, IBPSO achieves the best optimization results. Thus, we use these parameter values for all the datasets.

| $c_1$ | $c_2$ | Accuracy | Features | Fitness |
|-------|-------|----------|----------|---------|
| 1     | 1     | 0.9612   | 3.80     | 2.83    |
| 1     | 1.25  | 0.9613   | 4.33     | 3.17    |
| 1     | 1.5   | 0.9613   | 4.30     | 2.93    |
| 1     | 1.75  | 0.9613   | 4.30     | 2.93    |
| 2     | 1.75  | 0.9613   | 4.30     | 2.93    |

TABLE 2. Parameter tuning results of $c_1$ and $c_2$ in IBPSO.

3) EXPERIMENT SETUPS

In the feature selection tests, the genetic algorithm (GA), binary firefly algorithm (BFA), binary cuckoo search (BCS), BPSO, and binary bat algorithm (BBA) are introduced as the comparison algorithms. Moreover, the key parameter setups of these comparison algorithms as well as the proposed IBPSO are listed in Table 3. In addition, the maximum number of iterations for each algorithm is set as 200, the population size (number of searching agents) is 20, and the dimension of solution is equal to the feature...
TABLE 4. Fitness values obtained by different algorithms (Datasets 1 to 8).

| Dataset | Algorithm | Best | Worst | SD | Mean | CPU Time |
|---------|-----------|------|-------|----|------|----------|
| Breastcancer | GA | 0.0272 | 0.0272 | 0.0000 | 0.0272 | 69.9977 |
| | BFA | 0.0272 | 0.0309 | 0.0013 | 0.0288 | 138.5477 |
| | BCS | 0.0272 | 0.0272 | 0.0000 | 0.0272 | 20.8977 |
| | BPSO | 0.0272 | 0.0272 | 0.0000 | 0.0272 | 136.9305 |
| | BBA | 0.0272 | 0.0272 | 0.0000 | 0.0272 | 142.7571 |
| | IBPSO | 0.0272 | 0.0272 | 0.0000 | 0.0272 | 146.9765 |
| BreastEW | GA | 0.0389 | 0.0416 | 0.0008 | 0.0396 | 83.9037 |
| | BFA | 0.0399 | 0.0412 | 0.0004 | 0.0405 | 168.6450 |
| | BCS | 0.0433 | 0.0470 | 0.0010 | 0.0448 | 179.9466 |
| | BPSO | 0.0399 | 0.0409 | 0.0003 | 0.0404 | 171.6200 |
| | BBA | 0.0392 | 0.0399 | 0.0002 | 0.0396 | 167.6903 |
| | IBPSO | 0.0389 | 0.0392 | 0.0002 | 0.0390 | 168.3109 |
| Exactly | GA | 0.0046 | 0.0046 | 0.0000 | 0.0046 | 100.7601 |
| | BFA | 0.0046 | 0.0173 | 0.0038 | 0.0059 | 188.3626 |
| | BCS | 0.0046 | 0.0123 | 0.0028 | 0.0086 | 113.8868 |
| | BPSO | 0.0046 | 0.0046 | 0.0000 | 0.0046 | 198.0611 |
| | BBA | 0.0046 | 0.0046 | 0.0000 | 0.0046 | 199.2861 |
| | IBPSO | 0.0046 | 0.0046 | 0.0000 | 0.0046 | 197.6374 |
| Exactly2 | GA | 0.2097 | 0.2097 | 0.0000 | 0.2097 | 97.3038 |
| | BFA | 0.2097 | 0.2097 | 0.0000 | 0.2097 | 178.1391 |
| | BCS | 0.2097 | 0.2097 | 0.0000 | 0.2097 | 117.0541 |
| | BPSO | 0.2097 | 0.2097 | 0.0000 | 0.2097 | 186.8110 |
| | BBA | 0.2097 | 0.2097 | 0.0000 | 0.2097 | 177.7468 |
| | IBPSO | 0.2097 | 0.2097 | 0.0000 | 0.2097 | 184.6734 |
| HeartEW | GA | 0.1409 | 0.1648 | 0.0061 | 0.1529 | 57.3660 |
| | BFA | 0.1409 | 0.1607 | 0.0045 | 0.1525 | 113.1633 |
| | BCS | 0.1409 | 0.1638 | 0.0051 | 0.1572 | 62.2022 |
| | BPSO | 0.1409 | 0.1490 | 0.0000 | 0.1490 | 114.3146 |
| | BBA | 0.1409 | 0.1490 | 0.0000 | 0.1490 | 114.6943 |
| | IBPSO | 0.1409 | 0.1490 | 0.0000 | 0.1490 | 117.7609 |
| Lymphography | GA | 0.5385 | 0.5680 | 0.0065 | 0.5602 | 54.3351 |
| | BFA | 0.5385 | 0.5680 | 0.0065 | 0.5602 | 108.4117 |
| | BCS | 0.5385 | 0.5789 | 0.0063 | 0.5703 | 103.0773 |
| | BPSO | 0.5385 | 0.5789 | 0.0063 | 0.5703 | 108.7586 |
| | BBA | 0.5385 | 0.5704 | 0.0100 | 0.5558 | 108.5527 |
| | IBPSO | 0.5385 | 0.5704 | 0.0094 | 0.5551 | 112.4991 |
| SonarEW | GA | 0.0754 | 0.1075 | 0.0089 | 0.0938 | 69.3591 |
| | BFA | 0.0750 | 0.1038 | 0.0076 | 0.0947 | 138.3002 |
| | BCS | 0.0750 | 0.1099 | 0.0041 | 0.1151 | 145.1466 |
| | BPSO | 0.0946 | 0.1035 | 0.0029 | 0.0994 | 142.5023 |
| | BBA | 0.0754 | 0.0993 | 0.0066 | 0.0849 | 141.3256 |
| | IBPSO | 0.0547 | 0.0929 | 0.0113 | 0.0749 | 148.5394 |
| SpectEW | GA | 0.2534 | 0.2846 | 0.0100 | 0.2644 | 57.9660 |
| | BFA | 0.2603 | 0.2763 | 0.0055 | 0.2700 | 114.3705 |
| | BCS | 0.2589 | 0.2850 | 0.0080 | 0.2751 | 116.4804 |
| | BPSO | 0.2585 | 0.2731 | 0.0048 | 0.2661 | 116.6135 |
| | BBA | 0.2488 | 0.2662 | 0.0053 | 0.2575 | 116.1148 |
| | IBPSO | 0.2507 | 0.2777 | 0.0079 | 0.2623 | 119.6393 |

number of each dataset. Note that the performance of a metaheuristics algorithm is directly affected by the population size and the number of iterations. Specifically, if the algorithm is with large population size, it may achieve better optimization performance than the algorithm with small population size. Moreover, if an algorithm has more numbers of iterations, then it may obtain better results than the algorithm with less numbers of iterations. Thus, we use the same population size and the number of iteration for each algorithm to make a fair comparison between different algorithms.

Each algorithm is independently run for 30 times to solve the feature selection problems of these selected datasets, and the numerical statistics results will be presented. Moreover, in each test, we use 80 % of the instances for training, and the rest ones are used for testing, which is a common way adopted by several previous works.

The computer used for the tests is with an Intel(R) Xeon(R) E5-2630 v4 CPU and the RAM is 32 GB. Moreover, the abovementioned algorithms for feature selections are implemented by Python.

B. FEATURE SELECTION RESULTS

In this section, the feature selection results achieved by different algorithms are presented.

1) PERFORMANCE EVALUATIONS OF DIFFERENT ALGORITHMS

In this section, the fitness function values obtained by different algorithms are presented to show the performances of these approaches directly. Tables 4 and 5 show the numerical statistics results in terms of best value, worst value, standard deviation (SD), average value and CPU time of different algorithms for each datasets, and the best values obtained by
TABLE 5. Fitness values obtained by different algorithms (Datasets 9 to 16).

| Dataset   | Algorithm | Best     | Worst    | SD      | Mean    | CPU Time |
|-----------|-----------|----------|----------|---------|---------|----------|
| CongressEW| GA 0.0224 | 0.0279   | 0.0017   | 0.0236  | 75.2692 |
|           | BFA 0.0224| 0.0285   | 0.0016   | 0.0252  | 146.5245|
|           | BCS 0.0269| 0.0320   | 0.0016   | 0.0289  | 132.6274|
|           | IBPSO 0.0224| 0.0256 | 0.0013   | 0.0237  | 48.4214 |
|           | BBA 0.0224| 0.0281   | 0.0023   | 0.0249  | 48.0824 |
|           | IBPSO 0.0224| 0.0275 | 0.0020   | 0.0243  | 151.8781|
| IonosphereEW| GA 0.0846| 0.1141   | 0.0105   | 0.0997  | 78.1663 |
|           | BFA 0.0901| 0.1154   | 0.0070   | 0.1052  | 154.8571|
|           | BCS 0.1023| 0.1384   | 0.0095   | 0.1270  | 167.1765|
|           | IBPSO 0.0961| 0.1132 | 0.0059   | 0.1050  | 157.9756|
|           | BBA 0.0791| 0.1066   | 0.0088   | 0.0943  | 153.9433|
|           | IBPSO 0.0705| 0.1147 | 0.0140   | 0.0905  | 159.0922|
| KruskalEW| GA 0.0238 | 0.0359   | 0.0034   | 0.0295  | 882.3569|
|           | BFA 0.0338| 0.0454   | 0.0032   | 0.0389  | 1617.9190|
|           | BCS 0.0324| 0.0403   | 0.0023   | 0.0379  | 2082.1176|
|           | IBPSO 0.0276| 0.0371 | 0.0026   | 0.0322  | 1915.5977|
|           | BBA 0.0259| 0.0355   | 0.0029   | 0.0286  | 1982.8664|
|           | IBPSO 0.0210| 0.0276 | 0.0024   | 0.0241  | 2116.3775|
| Tic-tac-toe| GA 0.1564| 0.1564   | 0.0000   | 0.1564  | 89.3500 |
|           | BFA 0.1564| 0.1564   | 0.0000   | 0.1564  | 188.8746|
|           | BCS 0.1564| 0.1564   | 0.0000   | 0.1564  | 177.2706|
|           | IBPSO 0.1564| 0.1564 | 0.0000   | 0.1564  | 196.1019|
|           | BBA 0.1564| 0.1564   | 0.0000   | 0.1564  | 204.5367|
| Vote      | GA 0.0493 | 0.0630   | 0.0033   | 0.0558  | 58.6129 |
|           | BFA 0.0493| 0.0590   | 0.0028   | 0.0535  | 115.3932|
|           | BCS 0.0551| 0.0611   | 0.0024   | 0.0579  | 102.7837|
|           | IBPSO 0.0493| 0.0539 | 0.0017   | 0.0503  | 117.6988|
| WaveformEW| GA 0.1777| 0.1761   | 0.0054   | 0.1677  | 3739.2315|
|           | BFA 0.1777| 0.1834   | 0.0100   | 0.1810  | 6207.9594|
|           | BCS 0.1715| 0.1809   | 0.0027   | 0.1764  | 9771.4048|
|           | IBPSO 0.1644| 0.1769 | 0.0037   | 0.1721  | 7674.0827|
|           | BBA 0.1625| 0.1733   | 0.0035   | 0.1669  | 7524.7088|
|           | IBPSO 0.1573| 0.1698 | 0.0033   | 0.1621  | 7356.3544|
| Zoo       | GA 0.0584 | 0.0944   | 0.0107   | 0.0702  | 53.8852 |
|           | BFA 0.0584| 0.0757   | 0.0060   | 0.0639  | 108.6919|
|           | BCS 0.0584| 0.0764   | 0.0055   | 0.0668  | 89.2766 |
|           | IBPSO 0.0584| 0.0674 | 0.0027   | 0.0593  | 105.4454|
|           | BBA 0.0584| 0.0680   | 0.0028   | 0.0596  | 106.7008|
|           | IBPSO 0.0584| 0.0668 | 0.0025   | 0.0593  | 107.3102|
| Lungcancer| GA 0.0264| 0.0283   | 0.0006   | 0.0275  | 49.9007 |
|           | BFA 0.0271| 0.0281   | 0.0003   | 0.0277  | 102.6213|
|           | BCS 0.0281| 0.0549   | 0.0113   | 0.0373  | 100.4391|
|           | IBPSO 0.0287| 0.0287 | 0.0004   | 0.0280  | 103.4912|
|           | BBA 0.0269| 0.0280   | 0.0003   | 0.0276  | 102.6201|
|           | IBPSO 0.0256| 0.0269 | 0.0004   | 0.0264  | 107.6533|

a certain algorithm are highlighted in bold font for a clear presentation. Due to the limited page margin, the results of the selected 16 datasets are shown in two separated tables. It can be seen from the tables that the proposed IBPSO algorithm achieves the best average fitness function values on 12 datasets, which means it has better performance than other comparison algorithms. Moreover, IBPSO consumes more CPU time for solving the feature selection problems on most datasets. This is because the introduced improved factors need extra operations of solutions, which will cause the increasing the experiment time.

In addition, the convergence rates of different algorithms during the processes of solving the fitness functions are shown in Fig. 2. Note that these curves are selected from the 15th test, which is a median. As can be seen, the proposed IBPSO expose best curves on 12 datasets, which performs better convergence ability.

2) FEATURE SELECTION ACCURACIES
The feature selection accuracies obtained by different algorithm are presented in Tables 6 and 7, respectively. Similarly, the numerical statistics results of different algorithms for each datasets are presented in these tables. As can be seen, IBPSO algorithm achieves the best average accuracies of feature selection results on 10 datasets and the best accuracy results on 13 datasets. Thus, IBPSO algorithm has the best performance in terms of feature selection accuracies on these selected datasets compared to other algorithms. The reasons may be that the introduced improved factors can balance the exploration and exploitation.
abilities, so that enhancing the performance of the algorithm.

3) NUMBER OF SELECTED FEATURES
Tables 8 and 9 show the numbers of the selected features of the datasets obtained by different algorithms, respectively. Similar to the accuracy results, these tables also present the numerical statistics results. It can be seen from the tables that IBPSO obtains the best average number of selected features in half of the datasets (8 of 16), which can be regarded as the best results in the tests compared to other algorithms. Note that the accuracy and the 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, we may say that the proposed IBPSO has the overall best performance for feature selections in the selected datasets compared to other algorithms.

C. SOLUTION DISTRIBUTIONS
In this section, the exploration and exploitation changes of the proposed IBPSO are plotted such that the performance of the algorithm can be shown in a more intuitive way. However, the feature selection problem is usually with high solution dimensions, which means that it is very difficult to show the solution distributions directly. Thus, we use a popular data dimensionality reduction technology called t-SNE to visualize the high-dimensional data by giving each datapoint solution) in a two-dimensional map. The details and principle of t-SNE can be found in [57]. Without loss of generality and to make it more concise, we select four of sixteen datasets to present the visualization results of exploration and exploitation. The selected datasets are Breastcancer, CongressEW, BreastEW and SonarEW, since they are with different numbers of features (solution dimensions), which is representative. Moreover, the specific process of the population distribution visualization by using t-SNE is as follows. First, the solution distribution data of the population in each iteration are obtained and recorded. Second, these data are handled and mapped to the two-dimensional spaces by using t-SNE technology. Finally, we plot the mapped data of each selected dataset to show how exploration and exploitation are changed during the iterative process.

Fig. 3 shows the solution distributions in the population obtained by the proposed IBPSO. As can be seen,
the solutions obtained by IBPSO are with dense and sparse distributions alternately in the four selected datasets during the iterative processes, which means that IBPSO can balance the exploration and exploitation abilities in solving the feature selection problems.

D. EFFECTIVENESS OF THE IMPROVED FACTORS

In this section, we conduct tests to evaluate the effectiveness of the improved factors in IBPSO. In the test, we use conventional BPSO, BPSO with local search factor (BPSO-LSF), BPSO with global search factor (BPSO-GSF) and BPSO with population diversity improvement factor (BPSO-PDIF) to solve the formulated feature selection problem, respectively, to observe that whether these factors can enhance the performance of conventional PSO. The tests are also conducted on the four selected datasets that are Breastcancer, CongressEW, BreastEW and SonarEW, and the numerical results obtained by these abovementioned approaches are listed in Table 10. On the whole, all approaches achieve the same results on Breastcancer dataset and this may be that this dataset is with the lowest solution dimension, which makes it easy to be solved. The rest results are discussed in detail as follows.

1) LÉVY FLIGHT-BASED LOCAL SEARCH FACTOR

It can be seen from Table 10 that the fitness function value, classification accuracy and number of features obtained by BPSO-LSF are better than conventional BPSO algorithm. This is because the Lévy flight-based local search operator can extend the searching area of the global best solution, which means that it is able to provide a better exploitation
Table 8. Number of selected features obtained by different algorithms (Datasets 1 to 8).

| Dataset     | GA   | BPA  | BPSO | BCA  | IBPSO |
|-------------|------|------|------|------|-------|
| Breastcancer| 6.000| 6.000| 0.000| 6.000| 6.000 |
| BPA         | 3.000| 7.000| 1.2472| 5.6667|       |
| BPSO        | 6.000| 6.000| 0.000| 6.000| 6.000 |
| BCA         | 6.000| 6.000| 0.000| 6.000| 6.000 |
| IBPSO       | 6.000| 6.000| 0.000| 6.000| 6.000 |

Table 9. Number of selected features obtained by different algorithms (Datasets 9 to 16).

| Dataset     | GA   | BPA  | BPSO | BCA  | IBPSO |
|-------------|------|------|------|------|-------|
| CongressEW  | 4.000| 8.000| 1.4832| 6.0000|       |
| BPA         | 3.000| 11.000| 2.1190| 6.0000|       |
| BPSO        | 5.000| 8.000| 1.0770| 6.0000|       |
| BCA         | 7.000| 9.000| 1.7349| 6.0000|       |
| IBPSO       | 8.000| 1.2000| 6.0000|       |

2) WEIGHTING INERTIA COEFFICIENT-BASED GLOBAL SEARCH FACTOR

Table 10 shows that the accuracies of fitness function values obtained by BPSO-GSF performs better than BPSO, especially on medium-dimensional datasets. This is because that the proposed weighting inertia coefficient mechanism can adjust the searching scope of the algorithm adaptively such that improving the exploration ability. Note that this improved factor may decline the exploitation capability of the algorithm because it excessively retains the solutions of the previous iteration at the end-stage, so that reducing the performance of the algorithm on the datasets with larger solution dimensions. Thus, it is necessary to introduce the third improved factor to overcome this shortcoming.
FIGURE 3. Solution distributions during the iterative processes of different datasets obtained by IBPSO. (a) Breastcancer. (b) CongressEW. (c) BreastEW. (d) SonarEW.

3) MUTATION MECHANISM-BASED POPULATION DIVERSITY IMPROVEMENT FACTOR

As can be seen, the BPSO-PDIF approach can effectively achieve better fitness function value on SonarEW dataset, which is with the largest solution dimension. Moreover, Fig. 4 shows the convergence rates of BPSO and BPSO-PDIF on the four selected datasets. It can be seen from the figure that the proposed BPSO-PDIF scheme can improve the convergence ability compared to conventional BPSO algorithm. This is because that the mutation mechanism-based population diversity improvement factor makes the algorithm tend to exploit around high-quality solutions, so that enhancing the exploitation ability of the algorithm and convergence rate.

In summary, these three introduced improved factors are effective for improve the performance of conventional BPSO. Moreover, they are also complementary. For example, if the mutation mechanism is used on small size datasets, then the algorithm may easy to fall into local optima. Thus, it is necessary to adopt the Lévy flight-based local search factor to handle this issue.

E. LIMITATIONS OF IBPSO

Although the proposed IBPSO performs better performance than some other algorithms for the selected datasets, it still has some limitations. We think that the main limitations of this algorithm are as follows. First, the number of parameters in IBPSO is relatively more than some other approaches such as the conventional PSO algorithm, since the introduced improved factors bring more parameters. Thus, it is difficult to tune all the parameters of IBPSO for different applications, e.g., for solving other binary optimization problems. Second, IBPSO takes more experiment time than other algorithms even if they have the same computational complexity, and this can be reflected in Tables 4 and 5. This is because that the introduced improved factors need extra computing time than conventional PSO algorithm. For example, IBPSO has a Lévy flight-based local search operator, thus the algorithm should take more CPU time for calculation in each loop. Moreover, the proposed mutation mechanism also leads the algorithm to spend more CPU time. Finally, the performance of IBPSO for solving the feature selection problems with lower solution dimensions is not so good, and this can be also observed in Tables 4 and 5. The reason may be that the introduced mutation mechanism-based population diversity improvement factor can divide the solutions into different parts and let the algorithm to handle them separately, which is suitable for the larger population size with higher dimension of solution.

VI. CONCLUSION

In this paper, the feature selection problem is investigated. First, a joint feature selection problem is formulated, and then we propose an efficient algorithm called IBPSO to solve the formulated problem. In IBPSO, we first introduce the Lévy flight mechanism to improve the local search performance of the algorithm. Second, a weighting inertia coefficient operator is proposed to enhance the global search ability. Moreover, we use the mutation mechanism to improve the population diversity of the algorithm. Finally, a binary method is adopted to make the continuous algorithm suitable for the binary feature selection problem. Experiments are conducted on several classical datasets for the evaluations of the proposed algorithm, and the results show that the overall performance of IBPSO outperforms GA, BFA, BCS, BPSO and BBA for solving the feature selection problem. In our future work, more test datasets will be considered to further evaluate the proposed algorithm.

REFERENCES

[1] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” Science, vol. 349, no. 6245, pp. 255–260, Jul. 2015.
S. Sayed, M. Nassef, A. Badr, and I. Farag, “A nested genetic algorithm for feature selection: A data perspective,” ACM Comput. Surv., vol. 50, no. 6, pp. 1–45, Jan. 2018.

M. B. Ji, D. Ji, J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, “Feature selection: A review,” Expert Syst. Appl., vol. 140, Feb. 2020, Art. no. 112898.

J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, “Feature selection: A review,” Expert Syst. Appl., vol. 140, Feb. 2020, Art. no. 112898.

M. A. M. Anter, Y. Wei, J. Su, Y. Yuan, B. Lei, G. Duan, W. Mai, X. Nong, and S. Mirjalili, “A new fusion of grey wolf optimizer algorithm with a novel ACO–GA hybrid algorithm for feature selection in protein function prediction,” Expert Syst. Appl., vol. 7, pp. 670–687, Nov. 2019.

K. Mistry, L. Zhang, S. C. Neoh, C. P. Lim, and B. Fielding, “A micro-GA embedded PSO feature selection approach to intelligent facial emotion recognition,” IEEE Trans. Cybern., vol. 47, no. 6, pp. 1496–1509, Jun. 2017.

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.

M. Taradeh, M. Mafarja, A. A. Heidari, H. Faris, I. Aljarah, S. Mirjalili, and H. Fujita, “An evolutionary gravitational search-based feature selection,” Inf. Sci., vol. 497, pp. 219–239, Sep. 2019.

E. Emary, H. M. Zawbaa, K. K. A. Ghany, A. E. Hassanien, and B. Parv, “Firefly optimization algorithm for feature selection,” in Proc. 7th Balkan Conf. Inform. Conf. (BCI), 2015, pp. 1–7.

F. Tan, X. Fu, Y. Zhang, and A. G. Bourgeois, “A genetic algorithm-based method for feature subset selection,” Soft Comput., vol. 12, no. 2, pp. 111–120, 2008.

S. Sayed, M. Nassef, A. Badr, and I. Farag, “A nested genetic algorithm for feature selection in high-dimensional cancer microarray datasets,” Expert Syst. Appl., vol. 121, pp. 233–243, May 2019.

B. Ji et al.: Bio-Inspired Feature Selection: An IBPSO Approach.
[46] C. Qi, Z. Zhou, Y. Sun, H. Song, L. Hu, and Q. Wang, “Feature selection and multiple kernel boosting framework based on PSO with mutation mechanism for hyperspectral classification,” Neurocomputing, vol. 220, pp. 181–190, Jan. 2017.

[47] R. R. Chakrara, P. Sharma, and L. Singh, “A hybrid feature selection approach based on improved PSO and filter approaches for image steganalysis,” Int. J. Mach. Learn. Cybern., vol. 7, no. 6, pp. 1195–1206, Dec. 2016.

[48] S. Yadav, A. Ekbal, and S. Saha, “Information theoretic-PSO-based feature selection: An application in biomedical entity extraction,” Knowl. Inf. Syst., vol. 60, no. 3, pp. 1453–1478, Sep. 2019.

[49] K. Chen, F.-Y. Zhou, and X.-F. Yuan, “Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection,” Expert Syst. Appl., vol. 128, pp. 140–156, Aug. 2019.

[50] Y. Ding, K. Zhou, and W. Bi, “Feature selection based on hybridization of genetic algorithm and competitive swarm optimizer,” Soft Comput., pp. 1–10, Feb. 2020.

[51] S. B. Sakri, N. B. A. Rashid, and Z. M. Zain, “Particle swarm optimization feature selection for breast cancer recurrence prediction,” IEEE Access, vol. 6, pp. 29637–29647, 2018.

[52] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, “Efficient kNN classification algorithm for big data,” Neurocomputing, vol. 195, pp. 143–148, Jun. 2016.

[53] X.-S. Yang and S. Deb, “Cuckoo search via Lévy flights,” in Proc. World Congr. Nature Biologically Inspired Comput. (NaBIC), Dec. 2009, pp. 210–214.

[54] M. Abdel-Basset, D. El-Shahat, and A. K. Sangaiah, “A modified nature inspired meta-heuristic whale optimization algorithm for solving 0–1 knapsack problem,” Int. J. Mach. Learn. Cybern., vol. 10, no. 3, pp. 495–514, Mar. 2019.

[55] C. L. Blake and C. J. Merz, “UCI repository of machine learning databases,” Univ. California, Irvine, Irvine, CA, USA, Tech. Rep., 1998.

[56] 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.

[57] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol. 9, pp. 2579–2605, Nov. 2008.

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 the Chief Surgeon with Jilin University. His research interests include liver cancer and laparoscopic therapy.

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, feature engineering, and optimization.

JIAOHUI LU received the B.S. degree from the School of Clinical Medicine, Jilin University, in 2019. He is currently pursuing the degree with The First Hospital, Jilin University. His research interest includes machine learning in liver cancer.

YINZHE XIAO received the B.S. degree in naval architecture and ocean engineering from the Dalian University of Technology, in 2019. He is currently pursuing the master’s degree in computer science and technology with Jilin University. His research interests include data mining, feature engineering, and machine learning.

**B. Ji et al.: Bio-Inspired Feature Selection: An IBPSO Approach**

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, NY, and the University of Minnesota, in 2013 and 2016, respectively. She is currently a Professor with Jilin University. Her research interests include machine learning in liver cancer, chemotherapy resistance, and target therapy.

XIAOZHENG LU received the B.S. degree from the School of Clinical Medicine, Jilin University, in 2019. He is currently pursuing the degree with The First Hospital, Jilin University. His research interest includes machine learning in liver cancer.