Received March 4, 2020, accepted March 31, 2020, date of publication April 6, 2020, date of current version April 22, 2020.
Digital Object Identifier 10.1109/ACCESS.2020.2985986

OEbBOA: A Novel Improved Binary Butterfly Optimization Approaches With Various Strategies for Feature Selection

BO ZHANG[1], XINKAI YANG[1], BIAO HU[1], ZHAOGENG LIU[1,3], AND ZHANSHAN LI[1,2,3]
[1]College of Software, Jilin University, Changchun 130012, China
[2]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[3]Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun 130012, China
Corresponding author: Zhanshan Li (zslizsli@163.com)

This work supported in part by the Jilin Province Natural Science Foundation under Grant 20180101043JC, and in part by the Jilin Province Development and Reform Commission under Grant 2019C053-9.

ABSTRACT Binary butterfly optimization approach (bBOA) is a recent high performing feature selection algorithm presented in 2018 which is based on the food foraging behavior of butterflies. This paper tries to improve the structure of the bBOA to enhance its classification accuracy, dimension reduction and reliability in feature selection task for who are interested in the fields of data mining and pattern recognition. The new initialization strategy and differential evolution strategy are applied to reduce the randomness of bBOA's initialization and local search process. Then, a new parameter is added to make the bBOA's transfer function more adaptive to the change of exploration and exploitation. Besides, evolution population dynamics (EPD) mechanism is employed as an extension of bBOA. The new method called optimization and extension of binary butterfly optimization approaches (OEbBOA) is tested with the K nearest neighbor classier in which twenty UCI datasets and seven recent algorithms are utilized to assess the performance of the OEbBOA algorithm. The experimental results and nonparametric Wilcoxons rank sum test confirm the efficiency of the proposed OEbBOA in maximizing classification accuracy while minimizing the number of features selected.

INDEX TERMS Feature selection, evolutionary computation, differential evolution, evolutionary population dynamics.

I. INTRODUCTION

Feature selection is a process of selecting some of the most effective features from the original features to reduce the data dimensions [1], [2], which is a key pre-processing step in machine learning, data mining and pattern recognition. Feature selection is usually used in real tasks. While removing irrelevant and redundant features, the most important features are extracted from the acquired datasets to reduce the difficulty of solution search. Depending on the selection strategies, present feature selection methods can be broadly categorized into three types: filter method, embedded method and wrapped method [3].

The filter method does not rely on any learning algorithm, but the evaluation of certain features to asses their importance. The embedded method integrates the feature selection mechanisms into the training process of learning model, and automatically selects features along with the training of model [4]. The wrapped method relies on a predefined learning algorithm to evaluate the fitness of selected features. There are two main steps in a typical wrapped method: search strategy and sub-solution assessment. The difficulty of feature selection is that the search space will grow exponentially with the increasing of features [5]. Therefore, what search strategy to choose is the key to solve the feature selection problem. Due to the high global search capability, evolutionary computation has gained more and more attention on the field of feature selection in recent years [6], [7]. These algorithms have the ability to exploit useful population information to find the optimal solution [8]. Some of these algorithms are binary grasshopper optimisation algorithm approaches for feature selection [9], whale optimization approaches for wrapper feature selection [10], hybrid whale optimization algorithm with simulated annealing for feature selection [11]

The associate editor coordinating the review of this manuscript and approving it for publication was Li Wang.

and hybrid binary bat enhanced particle swarm optimization algorithm for solving feature selection [12].

Butterfly optimization algorithm (BOA) [13] is a recent meta-heuristic algorithm which is inspired by the food foraging behavior of butterfly. It is known for the capability to solve global optimization problems and is therefore applied to different applications, such as node localization in wireless sensor networks [14], engineering design problems [15], autonomous vehicle [16] and feature selection problem [8]. Attracted by the excellent performance of BOA, various variants of it have been proposed, such as butterfly optimization algorithm with artificial bee colony for numerical optimization [17], hybrid optimisation algorithm based on butterfly optimisation algorithm and differential evolution [18] and improved butterfly optimization algorithm with chaos [19]. Among them, binary butterfly optimization approaches for feature selection (bBOA) [8] is recently proposed to solve feature selection problem.

In bBOA, each feature subset is presented as a butterfly and each butterfly has its own scent and sensory organ of fragrance. The fragrance is related to the health of butterfly which is determined by the butterfly's fitness and the number of iterations. The algorithm applies two strategies: for a butterfly, if it can perceive the one with the largest scent of the search space, it will move towards that one, otherwise it will move randomly. The former strategy is called global search strategy and the latter is called local search strategy. Compared to other algorithms, bBOA can select the optimal feature subset which maximizes the classification accuracy while minimizing the length of the subset.

However, bBOA still has some shortcomings. First, the bBOA algorithm initializes the position of each butterfly randomly, which cannot promote the search process after the initialization. Second, since the bBOA algorithm is a binary variant of BOA, the transfer function [20] is applied to map computation results from continuous to discrete. However, the transfer function only serves as a mapping function in bBOA, which has no ability to balance the exploitation and exploration of the algorithm according to the health of butterflies. Thirdly, in the local search strategy of bBOA, butterfly simply adopts a random way to change its position, which is considered to be inefficient. In addition, usually there are some butterflies that have low fitness and therefore have very small probability of finding the optimal solution. But it is unable for bBOA to replace them with new butterflies born around the current optimal butterfly. For the defects described above, OEbBOA is proposed as an improved algorithm based on bBOA. The experimental results show that the proposed OEbBOA is competitive, compared with other recent feature selection algorithms.

II. OVERVIEW OF BINARY BUTTERFLY OPTIMIZATION APPROACHES (bBOA)

In bBOA, butterflies are initialized randomly first. Each of them can produce fragrance that is related to the fitness and can be calculated by *Eq*.(1). In addition, the fragrance can be sensed by other butterflies nearby. If a butterfly can sense the fragrance comes from the best butterfly in the search space, it will move towards the best butterfly by *Eq*.(3) and this stage is named as global search phase. If cannot, it will make random strides as described by *Eq*.(4) and this stage is named as local search phase. The general updating process of a butterfly's position is described by *Eq*.(2).

The fragrance is formulated as:

$$pf_i = cI^a \tag{1}$$

where pf_i is the perceived magnitude of fragrance which is produced by the ith butterfly, c is the sensory modality, I is the stimulus intensity which is implemented as the fitness of the ith butterfly and a is the power exponent depended on modality.

The updating process of butterfly's position vector can be described as:

$$x_i(t+1) = x_i(t) + F_i(t+1) \tag{2}$$

where $x_i(t)$ and $F_i(t+1)$ are the solution vector and moving magnitude of the ith butterfly at iteration number t. The move towards the best butterfly of the ith butterfly can be described as:

$$F_i(t+1) = (r^2 \times g^* - x_i(t)) \times pf_i \tag{3}$$

where $F_i(t+1)$ represents the moving amount which is utilized by the ith butterfly to update its position, $x_i(t)$ is the solution vector x_i for ith butterfly in iteration number t, g^* indicates the best solution found among all the solutions at current iteration and r is a uniform random number in [0, 1].

The random search of the ith butterfly can be described as:

$$F_i(t+1) = (r^2 \times x_j(t) - x_k(t)) \times pf_i \tag{4}$$

where $x_j(t)$ and $x_k(t)$ are the jth and kth butterflies from the solution space at iteration t and r is a uniform random number in [0, 1].

The pseudo code of bBOA algorithm is represented by Algorithm 1.

III. OPTIMIZATION AND EXTENSION OF BINARY BUTTERFLY OPTIMIZATION APPROACHES (OEbBOA)

OEbBOA is a variant of bBOA which is improved and extended according to the defects mentioned in section I. First, it utilizes a new initialization strategy that adds or subtracts features by greedy strategy after the feature importance ranking is obtained. Then, the transfer function is improved, in which the fragrance of butterflies is added as a new parameter, to enable the transfer function to adjust the development and exploration ability of algorithm according to the number of iterations and the health level of butterflies. After that, differential evolution is utilized to enhance the global search strategy and guide the local search strategy to reduce the randomness of the algorithm. Finally, butterfly replacement mechanism is proposed to eliminate butterflies whose fitness is relatively small and replace them with new butterflies born

Algorithm 1 Pseudo Code of bBOA

Objective function $f(x)$, $X = (x_1, x_2, \ldots, x_d)$
Generate a population of n butterflies $x_i(i = 1, 2, 3 \ldots, n)$
Define sensor modality c, power exponent a and switch probability p
while stopping criteria and not met **do**
  for each butterfly bf in population **do**
    Calculate the fragrance for bf using $Eq.(1)$
  end for
  Find the best bf
  for each butterfly bf in population **do**
    Generate a random number range from [0, 1]
    if $rand < p$ **then**
      Move towards the best butterfly using $Eq.(2)$ and $Eq.(3)$
    else
      Move randomly using $Eq.(2)$ and $Eq.(4)$
    end if
    Calculate the value of transfer function using $Eq.(5)$ or $Eq.(6)$
    Squash the solution using $Eq.(7)$ or $Eq.(8)$
    Evaluate the new butterfly
    If the new butterfly is better, update it in the population
  end for
  Update the value of c
  Find the current global best butterfly
end while
Output the best solution found

around the global optimal butterfly. The butterfly replacement mechanism is proposed to accelerate the convergence of the algorithm.

A. THE NEW INITIALIZATION STRATEGY

The original algorithm bBOA only uses the random method to initialize the butterfly's position, which cannot generate some butterflies with the ability to effectively guide the entire search process. Based on this problem, a heuristic initialization strategy combined with greedy strategy is proposed.

In the new strategy, each feature of the butterfly is treated as a gene and a certain number of butterflies with only one feature selected are generated. Then, the evaluation function described in section IV-B is employed to calculate the fitness value of these butterflies. The fitness reflects the combined contribution of the corresponding features to the classification accuracy and the dimension reduction in the case of a fixed classifier. Then, the fitness is ranked in descending order to get the ranking of features, which is regarded as the heuristic experience to guide the following initialization process.

Half of the butterflies generated randomly as the same way as the original method are selected to participate the

0 : unselected feature
1 : selected feature

1	1	0	1	0	0	1

FIGURE 1. A sample of the vector.

following steps. Each feature is first added to the vector that is depicted by Fig.1 according to the feature ranking order, i.e. if the bit has been already marked as 1, it remains unchanged, otherwise it will be set to 1 if the butterfly's fitness increases after the feature is added. Similarly, each feature is removed from the vector obtained from above according to the feature reverse order, i.e. if the bit has been already marked as 0, it remains unchanged, otherwise it will be set to 0 if the butterfly's fitness increases after the feature is removed. These two processes are repeated over and over again regardless of whether there are features added or removed until all features have been examined, which is the result of the application of the greedy strategy. The pseudo code of the above two processes is given in Algorithm 2.

Algorithm 2 The New Initialization Strategy

Importance metrics R, $x_i(0)$
Rank the features according to the importance metrics R
Create two candidate feature subsets S_1 and S_2, where S_1 is a collection of importance metrics R from large to small and S_2 is a collection of importance metrics R from small to large
To the feature subset O_i of the butterfly $x_i(0)$
for each feature in S_1 **do**
  Get the feature f from the collection S_1
  if f is added to the feature subset O_i, the fitness increases **then**
    $O_i \leftarrow O_i \bigcup f$
  end if
end for
for each feature in S_2 **do**
  Get the feature f from the collection S_2
  if f is deleted from the feature subset O_i, the fitness increases **then**
    $O_i \leftarrow O_i - f$
  end if
end for
Output the new $x_i(0)$ corresponding to the O_i

B. THE NEW TRANSFER FUNCTION

Transfer function is often applied to the algorithm using velocity vector. Its main function is to map velocity vector to probability, and then determines the value of each bit of the solution vector, which can realize the conversion of a new solution vector from continuous to discrete. Sigmoid (S-shaped) transfer function and V-shaped transfer function [21], [22] are both utilized in the search work of the bBOA, which is represented by $Eq.(5)$ and $Eq.(6)$ respectively.

S-shaped transfer function [23] can be represented as:

$$S(F_i^k(t)) = \frac{1}{1+e^{-F_i^k(t)}} \tag{5}$$

V-shaped transfer function [24] can be represented as:

$$V(F_i^k(t)) = |\frac{\sqrt{\pi}}{2}\int_0^{\frac{\sqrt{\pi}}{2}F_i^k(t)} e^{-t^2}dt| \tag{6}$$

where $F_i^k(t)$ is the continuous-valued moving magnitude of the the ith butterfly in kth dimension at iteration t.

The result of the $Eq.(5)$ and $Eq.(6)$ is the probability that a butterfly change the value of it's kth bit. Then, $Eq.(7)$ and $Eq.(8)$ are utilized to actually realize the conversion from continuous to discrete. The conversion function of the S-shaped transfer function:

$$x_i^k(t+1) = \begin{cases} 0 & if \quad rand < S(F_i^k(t)) \\ 1 & if \quad rand \geq S(F_i^k(t)) \end{cases} \tag{7}$$

The conversion function of the V-shaped transfer function:

$$x_i^k(t+1) = \begin{cases} (x_i^k(t))^{-1} & if \quad rand < V(F_i^k(t)) \\ x_i^k(t) & if \quad rand \geq V(F_i^k(t)) \end{cases} \tag{8}$$

where $x_i^k(t+1)$ represents the new position of the ith butterfly in kth dimension at iteration t. $S(F_i^k(t))$ and $V(F_i^k(t))$ are the probability from $Eq.(5)$ and $Eq.(6)$ with the moving magnitude $F_i^k(t)$ of the ith butterfly in kth dimension at iteration t.

According to the bBOA algorithm, compared with the V-shaped transfer function, the S-shaped transfer function is more suitable for mapping continuous moving value of butterflies' position since the mapping is more smooth, as shown in Fig.2. So only the modification of the S-shaped transfer function will be discussed.

As shown in Fig.2, $T(F)$ indicates the result of the sigmoid transfer function and x is the moving magnitude.

The S-shaped transfer function of the bBOA only uses the moving magnitude of the butterfly as its main parameter, which cannot adapt the exploration and exploitation in an evolutionary way during the search process. The evolution way means that when the butterfly is in the early stage of search or the butterfly is not healthy, the probability of changing its position is high, and then along with the search process, the probability decreases, which is conducive to the convergence of search.

According to the bBOA algorithm, the fragrance of each butterfly gradually decreases with the increasing of the iterations. It includes the fitness of butterfly, which not only describes the change of iteration, but also describes the health level of the butterfly.

Therefore, the fragrance of butterfly is added to the S-shaped transfer function as another parameter in addition to the moving magnitude. For the new transfer function, the smaller the number of iterations and fitness value are, the greater the need and possibility for a butterfly to change its position is.

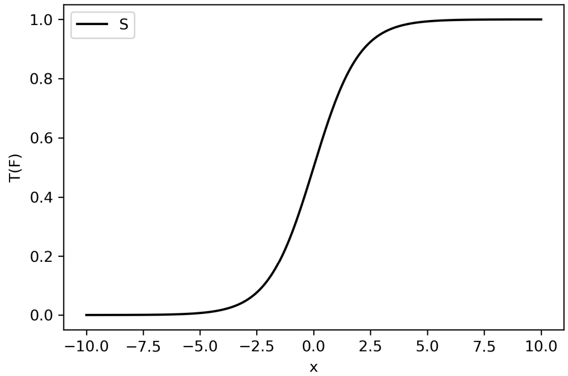


FIGURE 2. Sigmoid transfer function.

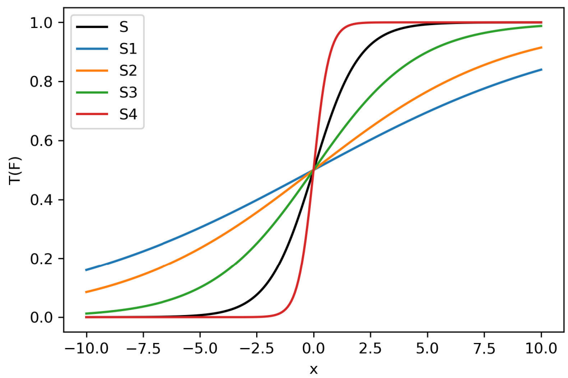


FIGURE 3. pf-varying S-shaped transfer function.

The new S-shaped transfer function which is named as pf-varying S-shaped transfer function is represented by $Eq.(7)$, (9) and (10), and its curve is demonstrated by Fig.3.

$$S(F_i^k(t), pf_i) = \frac{1}{1+e^{\frac{-F_i^k(t)}{\tau(pf_i)}}} \tag{9}$$

how τ changes is described as:

$$\tau(pf_i) = \frac{1}{c_{max}-c_{min}} \cdot pf_i \cdot \tau_{max} + (1 - \frac{1}{c_{max}-c_{min}} \cdot pf_i) \cdot \tau_{min} \tag{10}$$

where τ_{max} and τ_{min} are used to tune the bounds of the curve, which are determined by researchers. c_{max} and c_{min} are the upper and lower bounds of the fragrance. pf_i is the fragrance of the ith butterfly at iteration t, and the value of it is between $(0, c_{max} - c_{min})$.

Fig.3 shows how the probability described above varies with the magnitude of the movement x when τ_{max} is 4, τ_{min} is 0.01, and the total number of iterations is 100. S is the original S-shaped transfer function curve which is shown in Fig.2. S1, S2, S3 and S4 are the curves at the 25th, 50th, 75th, 100th iteration respectively. Because the fragrance of butterfly decreases with the increasing of iterations, Fig.3 also shows the relationship between the probability described above and the fragrance of butterfly.

C. DIFFERENTIAL EVOLUTION APPLIED TO SEARCH STRATEGIES

Differential evolution (DE) is a population-based meta-optimization algorithm proposed by Storn and Price in 1997 [25], which is known for its simplicity, effectiveness, and robustness. It is mainly divided into three steps: mutation, crossover and selection.

The mutation phase is used to generate a mutant vector [26]. In the mutation phase, D vectors different from the original vector are selected and used to replace the bits of original vector according to a fixed ratio called scaling factor [25] to obtain the mutant vector. In the crossover phase, the mutant vector obtained above and the original vector are combined and converted into a result vector called trial vector [26] according to a fixed ratio which is named as crossover rate [25]. Finally, in the selection phase, the fitness of the trial vector and its corresponding target vector which is generated by original method is evaluated and compared. The vector which has better performance will be chosen as a final vector.

These three differential evolution steps are often recycled to ensure that the final vector achieves the best result, but in the global and local search strategy of OEbBOA, each step is applied only once to prevent its effect from overlapping with other parts of the algorithm, resulting in additional consumption. In the global and local search of the OEbBOA, D is set to 2, i.e. the mutation vector is produced by two other butterflies and the original butterfly.

In the global search of the proposed method, for each butterfly, the original move method described by $Eq.(3)$ is used to calculate the temporary position of $x_{i,g}(t)$ in Algorithm 3, and differential evolution steps are employed to generate another temporary position. In global search phase, $Eq.(11)$ and $Eq.(12)$ are utilized to generate the mutant vector with the butterflies whose fitness are ranked first and second respectively.

$$v_i^k = x_i^k + F \cdot (x_{first}^k - x_{second}^k) \tag{11}$$

the conversion of the result from continuous to discrete is denoted as:

$$v_i^k = \begin{cases} 0 & v_i^k < 0.5 \\ 1 & otherwise \end{cases} \tag{12}$$

where v_i^k and x_i^k represent the mutant vector and current position of the ith butterfly in kth dimension, and F is a constant and real parameter within [0, 1]. The x_{first}^k and x_{second}^k are the kth dimensions of the butterflies whose fitness are ranked first and second respectively.

After the mutation stage, the trail vector is generated as the temporary position of the butterfly by $Eq.(13)$.

$$u_i^k = \begin{cases} v_i^k & rand_k[0,1] \leq CR \text{ or } k = k_{rand} \\ x_i^k & otherwise \end{cases} \tag{13}$$

where CR is a constant number which varies between [0, 1], and u_i^k and x_i^k denote the trail vector and the position of the ith butterfly in kth dimension. The result of $rand_k$ [0, 1] is a integer number within [0, 1], represented as k_{rand}. When k_{rand} equals k or lower than CR, the value of the u_i^k is set to v_i^k, otherwise it will be set to x_i^k.

In the final step of global search, the position with the best performance among the original position and the two temporary positions obtained above is selected as the real new position of butterfly.

Global search process of the OEbBOA is described by Algorithm 3.

Algorithm 3 New Global Search Strategy Utilizing Differential Evolution

1: $x_i(t)$, $x_{i,g}(t)$, $x_{first}(t)$, $x_{second}(t)$
2: Generate a mutant solution v_i by $Eq.(11)$ and $Eq.(12)$
3: Generate a trail solution u_i by $Eq.(13)$
4: Select the best vector among the u_i, $x_i(t)$ and $x_{i,g}(t)$ to generate the new butterfly $x_i(t+1)$
5: Output the $x_i(t+1)$

Similar to the global search, the differential evolution method is also used to guide butterflies in the local search stage of OEbBOA to reduce the blindness of the algorithm. In fact, the steps of the differential evolution applied to local search are the same as those applied to global search, except that one of the temporary positions, which is represented as $x_{i,l}(t)$ in Algorithm 4, is generated by the $Eq.(4)$ and the mutant vector is generated by the original butterfly and the other two other randomly selected butterflies through $Eq.(14)$ and $Eq.(12)$.

$$v_i^k = x_i^k + F \cdot (x_m^k - x_n^k) \tag{14}$$

where x_m^k and x_n^k are the kth dimensions of randomly selected butterflies, which are different from the ith butterfly. Other symbols have the same meaning as in $Eq.(11)$.

Local search process of the OEbBOA is described by Algorithm 4.

Algorithm 4 New Local Search Strategy Utilizing Differential Evolution

1: $x_i(t)$, $x_{i,l}(t)$, $x_m(t)$, $x_n(t)$
2: Generate a mutant solution v_i by $Eq.(14)$and $Eq.(12)$
3: Generate a trail solution u_i by $Eq.(13)$
4: Select the best vector among the u_i, $x_i(t)$ and $x_{i,l}(t)$ to generate a new butterfly $x_i(t+1)$
5: Output the $x_i(t+1)$

D. PROPOSED EVOLUTIONARY POPULATION DYNAMICS (EPD) STRATEGY IN OEbBOA

Evolutionary population dynamics(EPD) is the process to eliminate low-performance solutions in a population by relocating them around the best one [27]. This strategy can eliminate some solutions which are unlikely to find the optimal solution, and replace them with the solutions around the current best one according to a special mechanism [28], thereby accelerate the convergence of algorithm.

In the proposed approaches, the EPD strategy is specifically divided into two steps which are named as elimination and generation respectively. The elimination stage will delete half of the butterflies whose fitness ranking is in the second half. In the generation stage, three butterflies with the highest fitness and another different butterfly are selected to determine the bit value of each solution vector represented by the newly generated butterfly. To generate a new butterfly, for each bit of it, one of the four butterflies described above will be selected as the new value of the bit according to the same probability. Then, the process will then be repeated several times until all bits of the new butterfly are determined.

However, to prevent over fitting, each dimension of the newborn solutions will be set to the value which is opposite to the butterfly selected from the four butterflies described above by the probability of Pr. Pr is a constant number and is set to 0.7.

The pseudo code of the proposed EPD strategy is described by Algorithm 5.

Algorithm 5 Pseudo Code of EPD

$x_{first}(t)$, $x_{second}(t)$, $x_{third}(t)$, $x_{random}(t)$, $X = (x_1, \ldots, x_n)$
Get the ranking of the butterflies according to their fitness
Eliminate butterflies that account for half of the population by the converse order of the rank to get a population $X2 = (x_1, x_2, \ldots, x_{n/2})$
Create new butterflies of a number of half the population
for each new butterfly $x_i(t)$ **do**
  for each dimension k in the new butterfly **do**
    Select one butterfly $x_j(t)$ among $x_{first}(t)$, $x_{second}(t)$, $x_{third}(t)$ and $x_{random}(t)$ by probability of 0.25
    Set $x_i^k(t)$ value to the $x_j^k(t)$ or the opposite value according the probability Pr
  end for
  $X2 \leftarrow X2 \bigcup x_i(t)$
end for
Output the $X2$

In Algorithm 5, $x_{first}(t)$, $x_{second}(t)$ and $x_{third}(t)$ are the butterflies whose fitness ranking first, second and third respectively, and $x_{random}(t)$ is a randomly selected butterfly that is deferent from the three butterflies. $X2$ is the updated butterfly population using EPD.

To systematically show the improvements and extension, four sub versions of OEbBOA are proposed and realized to test their effectiveness and impact on improving the performance of bBOA. Then the conclusion is analyzed in detail in section IV-C based on the experimental results.

ObBOA_NIS: A method that only the new initialization strategy described in section III-A is included in. In this algorithm, by ranking the importance of individual features, some high-performance butterflies are generated to effectively guide the search process rather than initialize all butterflies randomly as in bBOA.

ObBOA_PFV: In this version, only the transfer function of bBOA is improved by adding the fragrance as a parameter in it, since the transfer function in bBOA only uses the moving magnitude of the butterfly as its main parameter and cannot adapt the exploration and exploitation by the increasing of iterations.

ObBOA_DE: This version has the differential evolution strategy described in section III-C applied in its search process. In its global search process, the new strategy can further prompt the performance of the algorithm by aggregating and reselecting excellent butterflies in an evolutionary way. Besides, in its local search process, the new strategy can guide the movement of butterflies instead of having them move randomly.

EbBOA_EPD: Only the EPD mechanism described in section III-D is employed in this version. The EPD is used to accelerate the convergence of the algorithm by eliminating the solutions which are unlikely to find the optimal solution and replace them with the solutions around the current best solution.

Finally, through the above four strategies, the proposed pseudo code of OEbBOA can be described by Algorithm 6.

IV. EXPERIMENTS AND RESULTS

A. DESIGN OF EXPERIMENTS

The K-nearest neighbor(KNN) is a simple and very common classifier that is widely used for wrapped method [29]. KNN classifier and twenty datasets which are shown in Table 1 are used to validate our proposed methods. All datasets come from the UCI machine learning repository [30].

To show the experimental results briefly, in following tables, 70%-30% means that the experiment is based on the datasets randomly split into 70% for training and 30% for testing. Similarly, 80%-20% means that the datasets are split into 80% for training and 20% for testing. Besides, in K-fold cross-validation, the dataset is divided into a number of folds where K-1 folds are utilized for training and rest folds are utilized for the testing purpose.

The proposed OEbBOA is realized by python 3.6 with open toolkit scikit-learn. And all experiments are carried out on a CPU i 5-3210 (Intel CoreTM Processor @2.50 GHz) computer.

The common parameters of OEbBOA and bBOA which are considered not to affect fairness are set to the same and the value of OEbBOA's own parameters, which are α, β, F, CR, τ_{max}, τ_{min} and Pr, is shown in Table 2.

Algorithm 6 Pseudo Code of OEbBOA

Objective function $f(x)$, $X = (x_1, x_2, \ldots, x_d)$
Generate a population with n butterflies $x_i(i = 1, 2, 3 \ldots, n)$
Update 50% butterflies by Algorithm 2
Define sensor modality c,power exponent a and switch probability p
while stopping criteria and not met **do**
  for each butterfly *bf* in population **do**
    Calculate the fragrance for *bf* using *Eq*.(1)
  end for
  Find the best *bf*
  for each butterfly *bf* in population **do**
    Generate a random number range from [0, 1]
    if *rand* $< p$ **then**
      Move towards the best butterfly using *Eq*.(2) and *Eq*.(3)
      Calculate the value of transfer function using *Eq*.(9) and *Eq*.(10)
      Figure out the temporary solution $x_{i,g}(t)$ using *Eq*.(7)
      Figure out the new butterfly $x_i(t+1)$ according to the Algorithm 3.
    else
      Move randomly using *Eq*.(2) and *Eq*.(4)
      Calculate the value of transfer function using *Eq*.(9) and *Eq*.(10)
      Figure out the temporary solution $x_{i,l}(t)$ using *Eq*.(7)
      Figure out the new butterfly $x_i(t+1)$ according to the Algorithm 4.
    end if
  end for
  Update 50% butterflies by EPD
  Update the value of c
  Find the current global best butterfly
end while
Output the best solution found

Seven recent feature selection algorithms are used to compare with the OEbBOA and their information is described in Table 3.

In order to ensure the reliability of the experimental results, all experimental results of the seven recently proposed algorithms described in Table 3 are from published papers. Most importantly, to fully test the effectiveness of the improvement, OEbBOA and the bBOA are compared in all the twenty datasets using the 5NN with all of their common parameters are the same except for the value of α and β in *Eq*.(17).

B. EVALUATION CRITERIA

Classification Accuracy(CA) and Dimension Reduction (DR) are used as the most basic evaluation indicators of the algorithm [31]–[33]. Their specific definitions are shown in *Eq*.(15) and *Eq*.(16), where N_CC (Number of Correct Classification) is the number of instances of the correct classification, N_AS (Number of All Samples) is the total number

TABLE 1. Specific information of the selected datasets.

S.no.	Dataset	Feature	Instance
1	Tae	5	151
2	Yeast	8	1484
3	Tic-tac-toe	9	958
4	Wine	13	178
5	Heart	13	270
6	Cleveland	13	303
7	Zoo	16	101
8	Lymphography	18	148
9	Vehicle	18	846
10	Segmentation	19	2310
11	Spect	22	267
12	Dermatology	33	366
13	Ionosphere	34	351
14	Waveform	40	5000
15	Spambase	57	4601
16	Sonar	60	208
17	LSVT	310	126
18	CNAE-9	856	1080
19	SRBCT	2308	63
20	Arcene	10000	200

TABLE 2. Parameter setting for experiments.

Parameter	Value(s)
M number of runs	20
P number of search agents	7
T number of iterations	100
Search domain	[0, 1]
a parameter in bBOA/OEbBOA	0.1
c parameter in bBOA/OEbBOA	[0.01, 0.25]
τ_{max} upper limit of the τ	4
τ_{min} lower limit of the τ	0.01
F scaling factor in OEbBOA	[0, 1]
CR crossover rate in OEbBOA	0.7
Pr random variation parameters in EPD	0.7

TABLE 3. Information of the methods for comparisons.

Algorithm	Description/Year of publication
bBOA	Binary butterfly optimization approaches for feature selection [9]/2019
PSO(4-2)	Particle Swarm Optimisation for feature selection [35]/2013
BGOA_M	Binary grasshopper optimisation algorithm approaches for feature selection [10]/2018
BGOA_EPD_Tour	Evolutionary Population Dynamics and Grasshopper Optimization approaches for feature selection problems [28]/2017
WOASAT-2	Hybrid Whale Optimization Algorithm with simulated annealing for feature selection [12]/2018
HBBEPSO	Hybrid Binary Bat Enhanced Particle Swarm Optimization Algorithm for solving feature selection [13]/2018
WOA-CM	Whale optimization approaches for wrapper feature selection [11]/2017
FSFOA	Feature selection using Forest Optimization Algorithm [32]/2016

TABLE 4. Average classification accuracy and dimension reduction of proposed approaches and bBOA.

S.no.	Dataset	OEbBOA		ObBOA_NIS		ObBOA_PFV		ObBOA_DE		EbBOA_EPD		bBOA	
		CA(%)	DR(%)	CA(%)	DR(%)	CA(%)	DR(%)	CA(%)	DR(%)	CA(%)	DR(%)	CA(%)	DR(%)
1	Tae	**60.04**	**62.21**	59.38	58.00	59.38	58.00	59.04	61.00	59.38	58.00	58.21	58.00
2	Yeast	**54.90**	**49.50**	54.44	47.50	54.44	47.50	54.52	46.25	54.44	47.50	53.64	46.88
3	Tic-tac-toe	**84.27**	50.37	83.42	48.15	83.04	48.89	81.88	55.56	83.62	48.89	79.83	**62.22**
4	Wine	**99.44**	**78.19**	99.39	76.38	99.33	72.31	99.30	73.36	99.28	73.39	98.43	52.30
5	Heart	**87.80**	64.39	87.42	**65.08**	86.73	62.61	86.71	64.46	87.41	64.16	82.37	55.38
6	Cleveland	**65.67**	**74.62**	63.98	73.08	64.14	71.54	64.29	67.69	64.32	69.24	64.32	73.44
7	Zoo	**98.11**	**68.24**	97.99	59.41	97.47	60.00	97.99	57.65	98.00	55.30	97.75	32.50
8	Lymphography	**90.73**	61.67	90.29	**62.11**	89.80	58.22	90.42	57.56	90.36	57.67	86.76	53.33
9	Vehicle	**79.43**	**61.44**	77.33	58.63	76.84	58.42	77.26	58.53	76.89	58.86	76.78	58.97
10	Segmentation	**95.92**	**74.74**	93.81	74.47	93.68	64.63	93.74	64.42	93.80	62.84	92.50	64.53
11	Spect	**85.16**	**63.24**	81.56	60.46	80.63	62.73	81.85	55.23	81.19	58.64	84.63	49.09
12	Dermatology	**100.00**	**72.37**	99.17	71.47	99.17	53.82	98.77	57.36	98.89	57.06	99.03	52.65
13	Ionosphere	**96.65**	**83.35**	95.28	78.63	94.10	64.59	94.18	66.08	94.06	65.94	90.70	52.35
14	Waveform	**84.10**	48.26	83.54	48.76	82.90	49.42	83.10	49.02	82.84	**49.50**	74.29	37.50
15	Spambase	**91.40**	**73.34**	90.81	68.77	90.15	57.20	90.17	57.02	89.94	60.36	90.06	58.60
16	Sonar	**95.14**	**81.70**	91.98	77.87	86.89	57.11	88.21	58.51	87.26	58.37	93.62	45.33
17	LSVT	**82.93**	**99.25**	79.05	99.13	71.90	55.37	71.75	55.61	73.91	55.62	71.83	55.31
18	CNAE-9	**93.72**	**89.39**	93.20	89.12	93.90	50.67	84.12	53.32	84.07	52.55	83.34	50.57
19	SRBCT	**100.00**	**99.64**	100.00	99.55	95.71	52.25	95.71	53.49	95.71	52.90	96.43	52.83
20	Arcene	**95.01**	**98.26**	94.52	97.89	90.01	50.73	89.49	51.96	89.49	52.00	89.12	52.06

of dataset instances, N_SF (Number of Selected Features) is the number of selected features, N_AF (Number of all features) is the total number of features in the dataset.

$$CA = N\_CC/N\_AS \cdot 100\% \quad (15)$$

$$DR = (1 - (N\_SF/N\_AF)) \cdot 100\% \quad (16)$$

In the OEbBOA, to decrease the feature subset length while increasing the classification accuracy, the fitness [35] in combination of CA and DR is used to evaluated solutions in the process of feature selection. The calculation of fitness relies on KNN classifier and is defined by $Eq.(17)$.

$$Fitness(x) = \alpha \cdot CA + \beta \cdot DR \quad (17)$$

where α and β are tune parameters, which is set to 0.9 and 0.1 in this work after a comparison between them and the combination of 0.99 and 0.01 [8]. However, since the setting of these two parameters is not considered to affect the fairness of experiment and the comparison is not in the scope of this study, the process has not been shown in this work.

Statistical standard deviation: represents the variation of the solutions obtained by executing an optimization algorithm for M times and can be formulated as:

$$Std.\ dev. = \sqrt{\frac{1}{M-1}\sum(g^{*i} - Mean)^2} \quad (18)$$

where g^{*i} represents the best solution in the ith run and $Mean$ denotes the average g^{*i} in total M runs [8].

C. EXPERIMENTAL RESULT AND ANALYSIS

1) COMPARISON AMONG PROPOSED APPROACHES AND bBOA

This subsection shows the results obtained by the five proposed approaches and the bBOA in terms of classification accuracy (CA), dimension reduction (DR), standard deviation, P-values of the Wilcoxon test and convergence curves of best fitness on twenty datasets. These approaches are compared to verify the effect of using new initialization strategy, new transfer function, differential evolution strategy and EPD mechanism. Experimental results are given in tables and figures while best value are presented in bold.

TABLE 5. Average fitness of proposed approaches and bBOA calculated by *Eq*.(17).

S.no.	Dataset	OEbBOA	ObBOA_NIS	ObBOA_PFV	ObBOA_DE	EbBOA_EPD	bBOA
1	Tae	**69.87**	69.87	69.87	69.87	69.87	69.87
2	Yeast	**57.25**	56.95	56.95	56.83	56.95	56.95
3	Tic-tac-toe	**86.34**	86.27	86.27	83.92	86.27	79.50
4	Wine	**97.32**	97.09	96.63	96.71	96.69	93.82
5	Heart	**85.46**	85.19	84.32	84.49	85.09	79.67
6	Cleveland	**69.09**	67.16	68.06	67.65	66.43	67.59
7	Zoo	**97.06**	97.06	96.18	96.77	96.18	96.47
8	Lymphography	**87.82**	87.47	86.64	87.13	87.09	83.42
9	Vehicle	**77.63**	75.46	75.00	75.39	75.09	75.00
10	Segmentation	**93.80**	91.88	90.78	90.81	90.70	89.70
11	Spect	**84.53**	83.00	82.64	84.22	82.86	82.52
12	dermatology	**98.24**	97.65	95.74	95.74	95.89	95.59
13	Ionosphere	**95.32**	93.62	91.15	91.37	91.25	86.87
14	Waveform	**80.52**	80.06	79.55	79.69	79.51	70.61
15	Spambase	**91.71**	91.56	89.25	89.26	88.84	89.47
16	Sonar	**93.80**	90.57	83.91	85.24	84.37	88.79
17	LSVT	**84.56**	81.06	70.25	70.14	72.08	70.18
18	CNAE-9	**93.29**	92.79	89.58	81.04	80.92	80.06
19	SRBCT	**99.99**	99.99	95.33	95.45	95.50	95.49
20	Arcene	**95.34**	94.86	86.08	85.74	85.74	85.41

From Table 4, It can be seen that all the proposed algorithms outperform bBOA on almost all datasets and OEbBOA performs best. For classification accuracy, OEbBOA can achieve better result than bBOA on all datasets and the difference between them varies from 1% to 11%. For instance, on Heart dataset, the OEbBOA's CA is 5% higher than bBOA's and on Waveform dataset, the OEbBOA's CA is 10% higher than bBOA's with 5NN classifier on 5-fold. Besides, other proposed algorithm, ObBOA_NIS, ObBOA_PFV, ObBOA_DE and EbBOA_EPD, also obtain higher value than bBOA on eleven datasets, especially Tic-tac-toe, Lymphography and Waveform. For dimension reduction, OEbBOA outperforms over other approaches on sixteen datasets, and over bBOA on nineteen datasets, like Zoo, on which OEbBOA's DR is 36% higher than bBOA's. For average fitness which is shown in Table 5, the best approach

TABLE 6. Comparison among proposed approaches and bBOA based on standard deviation which is defined by *Eq*.(18) using 5NN classifier on 5-fold.

S.no.	Dataset	OEbBOA	ObBOA_NIS	ObBOA_PFV	ObBOA_DE	EbBOA_EPD	bBOA
1	Tae	**0.001**	0.001	0.001	0.001	0.001	0.001
2	Yeast	**0.001**	0.001	0.001	0.208	0.001	0.001
3	Tic-tac-toe	2.457	**0.001**	0.001	0.443	0.001	0.811
4	Wine	0.124	**0.001**	0.001	0.385	0.385	0.050
5	Heart	**0.008**	0.450	1.090	0.450	0.450	0.008
6	Cleveland	0.670	0.495	1.135	**0.001**	1.010	2.170
7	Zoo	**0.001**	0.001	0.295	0.295	0.295	0.001
8	Lymphography	**0.013**	0.720	0.385	0.555	0.390	0.018
9	Vehicle	2.305	1.090	0.790	0.550	**0.450**	0.809
10	Segmentation	0.049	**0.001**	0.105	0.370	0.210	0.491
11	Spect	1.415	0.902	1.288	1.734	**0.574**	0.812
12	Dermatology	**0.001**	0.295	0.145	0.145	0.295	0.295
13	Ionosphere	**0.010**	1.286	0.584	0.380	0.365	0.010
14	Waveform	**0.001**	0.255	0.407	0.214	0.516	0.001
15	Spambase	0.355	0.440	0.040	**0.225**	0.260	0.090
16	Sonar	0.038	0.755	1.945	1.345	5.405	**0.001**
17	LSVT	0.029	**0.015**	0.145	0.210	0.865	0.092
18	CNAE-9	0.674	**0.130**	1.190	1.510	0.025	0.760
19	SRBCT	**0.001**	0.001	0.010	0.010	0.015	0.010
20	Arcene	0.662	0.001	**0.001**	0.001	0.001	0.732

TABLE 7. P-values of the Wilcoxon test of OEbBOA classification accuracy results vs other proposed algorithms ($p \geq 0.05$ are underlined).

S.no.	Dataset	OEbBOA	ObBOA_NIS	ObBOA_PFV	ObBOA_DE	EbBOA_EPD
1	Tae	<0.05	<0.05	<0.05	<0.05	<0.05
2	Yeast	<0.05	<0.05	<0.05	<0.05	<0.05
3	Tic-tac-toe	<0.05	<0.05	<0.05	<0.05	<0.05
4	Wine	<0.05	<0.05	<0.05	<0.05	<0.05
5	Heart	<0.05	<0.05	<0.05	<0.05	<0.05
6	Cleveland	<0.05	<0.05	<0.05	<0.05	0.096
7	Zoo	0.787	0.139	0.139	<0.05	<0.05
8	Lymphography	<0.05	<0.05	<0.05	<0.05	<0.05
9	Vehicle	<0.05	<0.05	<0.05	0.223	<0.05
10	Segmentation	<0.05	<0.05	<0.05	<0.05	<0.05
11	Spect	0.327	0.063	0.063	<0.05	<0.05
12	Dermatology	<0.05	<0.05	<0.05	<0.05	<0.05
13	Ionosphere	<0.05	<0.05	<0.05	<0.05	<0.05
14	Waveform	<0.05	<0.05	0.207	<0.05	0.114
15	Spambase	<0.05	<0.05	<0.05	0.361	<0.05
16	Sonar	<0.05	<0.05	<0.05	<0.05	<0.05
17	LSVT	<0.05	<0.05	<0.05	<0.05	<0.05
18	CNAE-9	<0.05	<0.05	<0.05	<0.05	<0.05
19	SRBCT	<0.05	<0.05	<0.05	<0.05	<0.05
20	Arcene	<0.05	<0.05	<0.05	<0.05	<0.05

is OEbBOA which outperform other approaches on nineteen out of twenty datasets and the difference between them varies from 1% to 14%. However, frankly speaking, not all the mechanisms can outperform over the bBOA in terms of average fitness such as Cleveland and Spambase. The results confirm the efficiency of these four mechanisms in improving the performance of bBOA according to classification accuracy and dimension reduction. It turns out that the new initialization strategy is more effective than other mechanisms in solving the datasets with relatively high dimensions.

To further discuss the promotion effect of four mechanisms when they aggregate into one algorithm, i.e., the OEbBOA, the nonparametric Wilcoxons rank sum test [9] is used to confirm whether there is a statistical difference among the proposed approaches and bBOA at 5% significance level. From Table 7, it can be seen that OEbBOA significantly outperforms them on most of the datasets, for which we can say that not only each of these four mechanisms has promotion effect, but also their facilitation can be effectively combined without being overwhelmed by any of them.

The robustness of proposed approaches is verified in Table 6, where it can be seen that OEbBOA has lower value of standard deviation on most datasets than other algorithms. It means that OEbBOA is more stable than other proposed methods and bBOA, such as Dermatology and Ionosphere, on which the standard deviation of OEbBOA is almost one order of magnitude lower than all other algorithms.

The average convergence performances of the proposed approaches with different mechanisms are demonstrated in Fig.4. From Fig.4, it can be seen that the convergence behavior of OEbBOA are more accelerated than other versions for more than half of the datasets, such as CNAE-9, LVST and Lymphography. However, the convergence process of ObBOA_DE suffers some kind of fluctuation, which means that the differential evolution mechanism is more suitable for the exploration rather than convergence.

To sum up, the mechanisms represented by ObBOA_NIS, ObBOA_PFV, ObBOA_DE and EbBOA_EPD all have promotion effect on the original algorithm bBOA on most of the datasets, and when they are aggregated into one algorithm, it can achieve the best results. However, there are some difference among these four mechanisms. The mechanism of ObBOA_NIS has the best promotion effect on classification accuracy and dimension reduction and its stability is second only to OEbBOA according to Table 4 and Table 6, but its convergence speed is lower than ObBOA_PFV and EbBOA_EPD and its time cost is relatively high according to section V. Although the mechanism of ObBOA_PFV has a limited effect on the improvement of various metrics, its negative effect on bBOA is the smallest, which proves that bBOA is more adaptable to it. ObBOA_DE mechanism's promotion to the classification and dimension reduction is remarkable, but it will lead to a longer exploration process and thus slows down the convergence of the algorithm. According to the Fig.4, the mechanism of ObBOA_EPD can significantly accelerate the convergence of the bBOA, like its performance on Arcene and Cleveland, but it would cause the decreasing of average fitness in some datasets, such as Cleveland and Spambase.

According to the discussion of time complexity, the new initialization strategy has the maximum time complexity and the change of transfer function has the least effect on execution time, which are indirectly verified by the performance of these strategies described above. Besides, through the analysis of the proposed algorithm's time complexity, it can be seen that the proposed algorithm has large time complexity, which may lead to long execution time.

2) COMPARISON WITH OTHER RECENT FEATURE SELECTION ALGORITHMS

This part presents the comparison between OEbBOA, as the best one among the proposed approaches, and other recent algorithms. The algorithms that are used for this comparison are PSO(4-2), BGOA_M, BGOA_EPD_Tour, WOASAT-2, HBBEPSO, WOA-CM and FSFOA, which are considered as high-performance and well-regarded algorithms. In order to ensure the reliability of the experiment, all comparison are based on published results using same classifier and dataset division method.

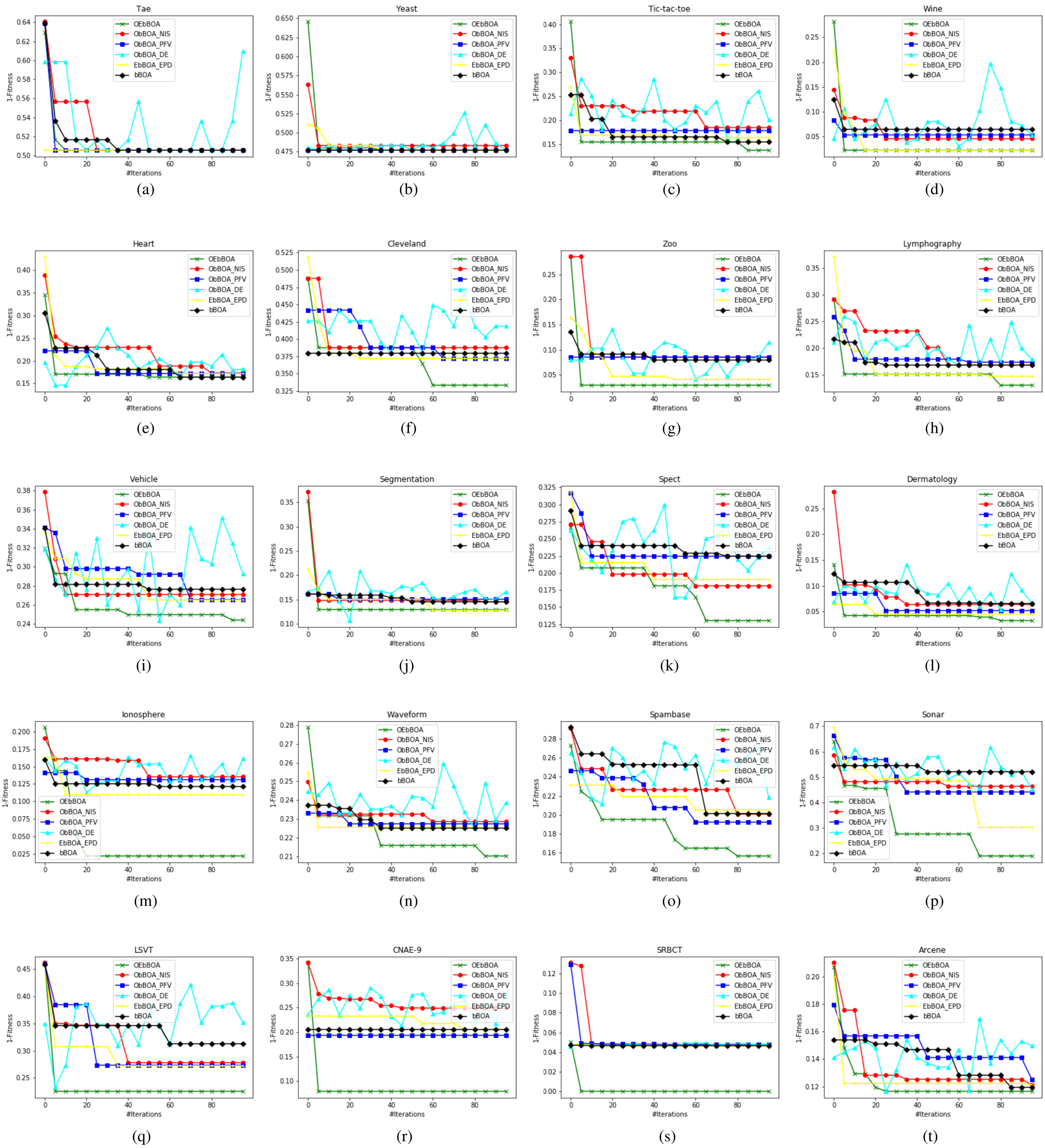


FIGURE 4. **Convergence curves of the proposed approaches and bBOA with the 5NN classifier on 5-fold.**

Table 8 to Table 27 outline the comparison in terms of classification accuracy and dimension reduction. For classification accuracy, OEbBOA can achieve better result than other algorithms on most of the datasets. For example, on Lymphography dataset, OEbBOA's CA is 6% higher than BGOA_EPD_Tou's with 5NN classifier on 5-fold. As the same as that, on Sonar datasets, OEbBOA's CA is 8% higher than FSFOA with 5NN classifier on 70%-30% method. For dimension reduction, OEbBOA outperforms over other algorithms for more than half of the datasets and the difference between them varies from 1% to 46%. For instance, OEbBOA's DR is 42% higher than FSFOA on Segmentation and 9% higher than HBBESPO on Sonar.

TABLE 8. The results of the proposed OEbBOA and its comparison algorithm on Tae.

Tae	CA(%)	DR(%)	Classifier
OEbBOA	**60.04(5-fold)**	**62.21**	5NN
bBOA	58.21(5-fold)	58.00	5NN

TABLE 9. The results of the proposed OEbBOA and its comparison algorithm on Yeast.

Yeast	CA(%)	DR(%)	Classifier
OEbBOA	**54.90(5-fold)**	**49.50**	5NN
bBOA	53.64(5-fold)	46.88	5NN

TABLE 10. The results of the proposed OEbBOA and its comparison algorithm on Tic-tac-toe.

Tic-tac-toe	CA(%)	DR(%)	Classifier
OEbBOA	**84.27(5-fold)**	50.37	5NN
bBOA	79.83(5-fold)	62.22	5NN
WOASAT	76.00(5-fold)	**66.67**	5NN
WOA-CM	78.54(5-fold)	23.30	5NN
OEbBOA	**76.60(2-fold)**	**60.19**	5NN
HBBEPSO	74.5(2-fold)	48.90	5NN
OEbBOA	**82.46(80%-20%)**	46.30	5NN
BGOA_M	80.38(80%-20%)	**66.67**	5NN
BGOA_EPD_Tour	80.80(80%-20%)	55.56	5NN

TABLE 11. The results of the proposed OEbBOA and its comparison algorithm on Cleveland.

Cleveland	CA(%)	DR(%)	Classifier
OEbBOA	**65.67(5-fold)**	**74.62**	5NN
bBOA	64.32(5-fold)	73.44	5NN
OEbBOA	**59.82(70%-30%)**	**71.79**	1NN
FSFOA	55.55(70%-30%)	71.42	1NN

TABLE 12. The results of the proposed OEbBOA and its comparison algorithm on Zoo.

Zoo	CA(%)	DR(%)	Classifier
OEbBOA	**98.11(5-fold)**	**68.24**	5NN
bBOA	97.75(5-fold)	32.50	5NN
WOASAT	97.00(5-fold)	35.00	5NN
WOA-CM	98.04(5-fold)	62.50	5NN
OEbBOA	**94.68(2-fold)**	65.20	5NN
HBBEPSO	87.30(2-fold)	**68.20**	5NN
OEbBOA	96.82(80%-20%)	**62.75**	5NN
BGOA_M	97.78(80%-20%)	33.75	5NN
BGOA_EPD_Tour	**99.30(80%-20%)**	56.25	5NN
OEbBOA	**96.77(70%-30%)**	**76.47**	1NN
PSO (4-2)	95.52(70%-30%)	40.38	1NN

TABLE 13. The results of the proposed OEbBOA and its comparison algorithm on Spect.

Spect	CA(%)	DR(%)	Classifier
OEbBOA	85.16(5-fold)	63.24	5NN
bBOA	84.63(5-fold)	49.09	5NN
WOASAT	84.00(5-fold)	42.72	5NN
WOA-CM	**86.57(5-fold)**	**63.40**	5NN
OEbBOA	**77.53(2-fold)**	**65.53**	5NN
HBBEPSO	76.4(2-fold)	57.60	5NN
OEbBOA	80.86(80%-20%)	**57.27**	5NN
BGOA_EPD_Tour	**82.60(80%-20%)**	50.00	5NN

As a whole, it can be seen that OEbBOA can outperform over other recent feature selection algorithms on most of the datasets in terms of classification accuracy and dimension reduction. Through the analysis of the proposed four strategies, the superiority comes from the high performance of bBOA itself and the improvements to its defects, in which bBOA is presented in 2018 and considered as a excellent algorithm.

TABLE 14. The results of the proposed OEbBOA and its comparison algorithm on Dermatology.

Dermatology	CA(%)	DR(%)	Classifier
OEbBOA	**100.00(5-fold)**	**72.37**	5NN
bBOA	99.03(5-fold)	52.65	5NN
OEbBOA	**98.36(2-fold)**	**65.94**	5NN
HBBEPSO	93.30(2-fold)	53.00	5NN
OEbBOA	**99.30(80%-20%)**	**69.12**	5NN
BGOA_M	99.23(80%-20%)	42.42	5NN
OEbBOA	**98.94(70%-30%)**	**67.64**	1NN
PSO (4-2)	97.27(70%-30%)	45.71	1NN

TABLE 15. The results of the proposed OEbBOA and its comparison algorithm on Spambase.

Spambase	CA(%)	DR(%)	Classifier
OEbBOA	**91.40(5-fold)**	**73.34**	5NN
bBOA	90.06(5-fold)	58.60	5NN
OEbBOA	92.53(80%-20%)	**70.18**	5NN
BGOA_M	**93.49(80%-20%)**	56.14	5NN

TABLE 16. The results of the proposed OEbBOA and its comparison algorithm on SRBCT.

SRBCT	CA(%)	DR(%)	Classifier
OEbBOA	**100.00(5-fold)**	**99.64**	5NN
bBOA	96.43(5-fold)	52.83	5NN
OEbBOA	**96.77(70%-30%)**	**76.47**	1NN
PSO (4-2)	94.73(70%-30%)	49.06	1NN

TABLE 17. The results of the proposed OEbBOA and its comparison algorithm on Heart.

Heart	CA(%)	DR(%)	Classifier
OEbBOA	**87.80(5-fold)**	**64.39**	5NN
bBOA	82.37(5-fold)	55.38	5NN
WOASAT-2	85.00(5-fold)	58.46	5NN
WOA-CM	80.67(5-fold)	46.50	5NN
OEbBOA	**86.72(2-fold)**	74.62	5NN
bBOA	85.05(2-fold)	65.38	5NN
HBBEPSO	81.30(2-fold)	**81.30**	5NN
OEbBOA	**89.33(80%-20%)**	**68.97**	5NN
bBOA	87.59(80%-20%)	64.10	5NN
BGOA_M	83.58 (80%-20%)	60.00	5NN
BGOA_EPD_Tour	83.30(80%-20%)	35.38	5NN

TABLE 18. The results of the proposed OEbBOA and its comparison algorithm on Segmentation.

Segmentation	CA(%)	DR(%)	Classifier
OEbBOA	**93.92(5-fold)**	**76.63**	5NN
bBOA	92.50(5-fold)	64.53	5NN
OEbBOA	**96.84(70%-30%)**	**78.95**	1NN
bBOA	95.79(70%-30%)	66.98	1NN
FSFOA	96.51(70%-30%)	36.84	1NN

V. COMPLEXITY ANALYSIS

Consider T as the total number of iterations of the algorithm, P is the number of butterflies, H is the number of instances,

TABLE 19. The results of the proposed OEbBOA and its comparison algorithm on Arcene.

Arcene	CA(%)	DR(%)	Classifier
OEbBOA	**95.01(5-fold)**	**98.26**	5NN
bBOA	89.12(5-fold)	52.06	5NN
OEbBOA	**96.67(70%-30%)**	**98.45**	1NN
bBOA	95.06(70%-30%)	52.41	1NN
FSFOA	88.33(70%-30%)	61.9	1NN

TABLE 20. The results of the proposed OEbBOA and its comparison algorithm on Sonar.

Sonar	CA(%)	DR(%)	Classifier
OEbBOA	95.14(5-fold)	**81.7**	5NN
bBOA	93.62(5-fold)	45.33	5NN
WOASAT-2	**97.00(5-fold)**	56.00	5NN
WOA-CM	91.88(5-fold)	59.40	5NN
OEbBOA	**90.48(70%-30%)**	69.87	5NN
bBOA	87.14(70%-30%)	56.53	5NN
PSO(4-2)	78.16(70%-30%)	**81.27**	5NN
FSFOA	86.98(70%-30%)	44.26	5NN
OEbBOA	**71.55(2-fold)**	**67.12**	5NN
bBOA	66.74(2-fold)	58.26	5NN
HBBESPO	70.40(2-fold)	58.70	5NN
OEbBOA	**93.97(80%-20%)**	**71.06**	5NN
bBOA	91.46(80%-20%)	57.5	5NN
BGOA_M	91.47(80%-20%)	55.33	5NN
BGOA_EPD_Tour	91.20 (80%-20%)	38.72	5NN
OEbBOA	**93.81(70%-30%)**	**68.75**	1NN
bBOA	92.78(70%-30%)	55.82	1NN
FSFOA	85.43(70%-30%)	57.37	1NN

TABLE 21. The results of the proposed OEbBOA and its comparison algorithm on Wine.

Wine	CA(%)	DR(%)	Classifier
OEbBOA	**99.44(5-fold)**	**78.14**	5NN
bBOA	98.43(5-fold)	52.30	5NN
WOASAT-2	99.00(5-fold)	50.77	5NN
WOA-CM	95.90(5-fold)	47.70	5NN
OEbBOA	98.30(70%-30%)	**69.23**	5NN
bBOA	97.04(70%-30%)	62.31	5NN
PSO(4-2)	95.26(70%-30%)	47.38	5NN
FSFOA	**99.20(70%-30%)**	30.76	5NN
OEbBOA	**97.61(2-fold)**	**68.45**	5NN
bBOA	95.56(2-fold)	59.19	5NN
HBBEPSO	90.10(2-fold)	60.80	5NN
OEbBOA	**99.17(80%-20%)**	**73.49**	5NN
bBOA	98.98(80%-20%)	70.26	5NN
BGOA_M	98.88(80%-20%)	66.15	5NN
BGOA_EPD_Tour	98.90(80%-20%)	32.30	5NN
OEbBOA	**98.15(70%-30%)**	**67.31**	1NN
bBOA	97.95(70%-30%)	62.69	1NN
FSFOA	98.07(70%-30%)	50.00	1NN

TABLE 22. The results of the proposed OEbBOA and its comparison algorithm on LSVT.

LSVT	CA(%)	DR(%)	Classifier
OEbBOA	**82.93(5-fold)**	**99.25**	5NN
bBOA	71.83(5-fold)	55.31	5NN
OEbBOA	75.39(70%-30%)	**99.11**	1NN
bBOA	53.82(70%-30%)	53.82	1NN
FSFOA	**89.47(70%-30%)**	98.71	1NN

C_0 is the time complexity of base classifier used in the OEbBOA. f_{num} is the number of features, γ is the ratio of non-zero importance features, and the number of non-zero importance features $m = f_{num} \cdot \gamma$.

TABLE 23. The results of the proposed OEbBOA and its comparison algorithm on Waveform.

Waveform	CA(%)	DR(%)	Classifier
OEbBOA	**84.1(5-fold)**	48.26	5NN
bBOA	74.29(5-fold)	37.5	5NN
WOASAT-2	76.00(5-fold)	**65.66**	5NN
WOA-CM	75.33(5-fold)	36.50	5NN
OEbBOA	83.13(70%-30%)	**55.36**	5NN
bBOA	81.17(70%-30%)	50.24	5NN
PSO(4-2)	**86.62(70%-30%)**	27.75	5NN
OEbBOA	**82.49(80%-20%)**	**53.11**	5NN
bBOA	80.94(80%-20%)	47.64	5NN
BGOA_M	75.11(80%-20%)	47.75	5NN
BGOA_EPD_Tour	73.70 (80%-20%)	34.41	5NN

TABLE 24. The results of the proposed OEbBOA and its comparison algorithm on Vehicle.

Vehicle	CA(%)	DR(%)	Classifier
OEbBOA	**79.43(5-fold)**	**61.44**	5NN
bBOA	76.78(5-fold)	58.97	5NN
OEbBOA	75.42(70%-30%)	**60.28**	5NN
bBOA	72.29(70%-30%)	59.72	5NN
PSO(4-2)	**85.30(70%-30%)**	43.56	5NN
FSFOA	73.98(70%-30%)	50.00	5NN
OEbBOA	**77.71(80%-20%)**	**63.33**	5NN
bBOA	74.55(80%-20%)	60.93	5NN
BGOA_M	77.04(80%-20%)	46.66	5NN
OEbBOA	73.63(70%-30%)	**62.44**	1NN
bBOA	71.82(70%-30%)	60.85	1NN
FSFOA	**73.81(70%-30%)**	59.78	1NN

TABLE 25. The results of the proposed OEbBOA and its comparison algorithm on Ionosphere.

Ionosphere	CA(%)	DR(%)	Classifier
OEbBOA	**96.65(5-fold)**	**83.35**	5NN
bBOA	90.7(5-fold)	52.35	5NN
WOASAT-2	96.00(5-fold)	62.35	5NN
WOA-CM	92.56(5-fold)	57.60	5NN
OEbBOA	**91.51(70%-30%)**	71.86	5NN
bBOA	89.43(70%-30%)	67.26	5NN
PSO(4-2)	87.27(70%-30%)	**90.41**	5NN
OEbBOA	**87.52(2-fold)**	**74.42**	5NN
bBOA	85.54(2-fold)	68.32	5NN
HBBEPSO	83.10(2-fold)	62.40	5NN
OEbBOA	92.83(80%-20%)	**70.20**	5NN
bBOA	90.19(80%-20%)	69.71	5NN
BGOA_M	**94.58 (80%-20%)**	66.27	5NN
BGOA_EPD_Tour	89.90(80%-20%)	51.76	5NN
OEbBOA	**93.96(70%-30%)**	**65.44**	1NN
bBOA	92.21(70%-30%)	61.32	1NN
FSFOA	89.52(70%-30%)	54.28	1NN

TABLE 26. The results of the proposed OEbBOA and its comparison algorithm on CNAE-9.

CNAE-9	CA(%)	DR(%)	Classifier
OEbBOA	**93.72(5-fold)**	**89.39**	5NN
bBOA	83.34(5-fold)	50.57	5NN
OEbBOA	**93.95(70%-30%)**	**90.64**	1NN
bBOA	83.27(70%-30%)	50.22	1NN
FSFOA	91.05(70%-30%)	24.88	1NN

First, to analyse the time complexity of the initialization strategy, the initialization process is divided into two parts: ranking and solution optimization. The time complexity of

TABLE 27. The results of the proposed OEbBOA and its comparison algorithm on Lymphography.

Lymphography	CA(%)	DR(%)	Classifier
OEbBOA	**90.73(5-fold)**	**61.67**	5NN
bBOA	86.76(5-fold)	53.33	5NN
WOASAT-2	89.90(5-fold)	60.00	5NN
OEbBOA	**86.42(2-fold)**	**65.50**	5NN
bBOA	84.35(2-fold)	61.94	5NN
HBBEPSO	42.20(2-fold)	64.50	5NN
OEbBOA	**92.67(80%-20%)**	**62.93**	5NN
bBOA	91.11(80%-20%)	57.59	5NN
BGOA_M	91.18 (80%-20%)	49.81	5NN
BGOA_EPD_Tour	86.80(80%-20%)	40.92	5NN

ranking the features's importance is $O(f_{num})\times C_0$. For solution optimization, the ranking list is sequentially checked by both ascending order and the descending order. Therefore, the time complexity of solution optimization of $P/2$ butterflies is $O(2\cdot f_{num}\cdot P/2)\times C_0$. After the initialization process, in one iteration of the algorithm, the time complexity of calculating the fragrance of each butterfly is $O(P)$. In global and local search phase, the time complexity required to calculate new positions of butterflies by the original method different from differential evolution is $O(P\cdot f_{num})+(O(P)\times C_0)$. In differential evolution strategy, the required time complexity is $O(P\cdot f_{num})+(O(P)\times C_0)$. Finally, the time complexity required to replace butterflies and update the fitness of $P/2$ butterflies is $O(f_{num}\cdot P/2)+(O(P/2)\times C_0)$. The kNN classifier regarded as a base classifier is utilized and is implemented based on Ball Tree. The time complexity of it is $O(mHlogH)$ and therefore C_0 equals $O(mHlogH)$. To sum up, the total time complexity of the OEbBOA algorithm can be approximated described as the formula described by $Eq.$(19):

$$O(f_{num}\cdot PmHlogH + PTmHlogH + f_{num}\cdot PT) \quad (19)$$

VI. CONCLUSION

In this work, based on binary butterfly optimization approach (bBOA), an improved algorithm for feature selection is proposed and named as OEbBOA. Four strategies are used to promote and extend the performance of bBOA in classification accuracy and dimension reduction. Twenty datasets from UCI are used and the experimental results are compared with seven recent high-performance feature selection algorithms such as binary grasshopper optimisation algorithm approaches for feature selection (BGOA_M), whale optimization approaches for wrapper feature selection (WOA_CM) and hybrid binary bat enhanced particle swarm optimization algorithm for solving feature selection (HBBEPSO). Through statistical analysis, the complementary effect of the four strategies on bBOA is verified like the facilitation of the new initialization strategy for datasets with relatively high dimensions and the exploration ability of the differential evolution strategy. It is observed that aggregating four strategies, OEbBOA has remarkable superiority over other algorithms. Besides, according to convergence curves, it also shows that the convergence speed of the proposed algorithm is higher than bBOA's.

However, there are still some shortcomings in the proposed approaches. First, although the exploration ability of ObBOA_DE is high, it would slow down the convergence speed of the whole algorithm. Then, since the significance of each feature is measured in the new initialization strategy, its time complexity is relatively high, resulting in the actual run time of OEbBOA is long. In addition, as the run time of the new initialization strategy increases with the increasing of the dataset dimensions, it is difficult for OEbBOA to solve the datasets with higher dimensions than 10000. Different from other mechanisms, the relationship between the parameters of EPD mechanism and those in bBOA is weak. Since the EPD mechanism is sensitive to the setting of its parameters, to ensure the efficiency of the EPD, the value of its parameters in this work is the same as those in cited papers [27].

Future studies can concentrate on the application of the new initialization strategy and the EPD strategy to other population-based optimizers. The efficacy of the proposed OEbBOA can also be employed to tackle other data mining problems. For future works, we intended to compare the proposed OEbBOA with different types of feature selection algorithms, and discuss the application of OEbBOA and its sub versions in other engineering fields. Besides, we will also focus on how to reduce the total time complexity of the OEbBOA to a more acceptable level and make the algorithm enable for the processing of the datasets with higher dimension like 100000 by utilizing the strategies in other algorithms such as CatBoost and LightGBM.

REFERENCES

[1] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005.
[2] 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.
[3] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, p. 94, 2018.
[4] E. Hüllermeier and M. Rifqi, "A fuzzy variant of the rand index for comparing clustering structures," in *Proc. IFSA/EUSFLAT Conf.*, 2009, pp. 1294–1298.
[5] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.
[6] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.
[7] X.-Y. Liu, Y. Liang, S. Wang, Z.-Y. Yang, and H.-S. Ye, "A hybrid genetic algorithm with wrapper-embedded approaches for feature selection," *IEEE Access*, vol. 6, pp. 22863–22874, 2018.
[8] S. Arora and P. Anand, "Binary butterfly optimization approaches for feature selection," *Expert Syst. Appl.*, vol. 116, pp. 147–160, Feb. 2019.
[9] M. Mafarja, I. Aljarah, H. Faris, A. I. Hammouri, A. M. Al-Zoubi, and S. Mirjalili, "Binary grasshopper optimisation algorithm approaches for feature selection problems," *Expert Syst. Appl.*, vol. 117, pp. 267–286, Mar. 2019.
[10] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput.*, vol. 62, pp. 441–453, Jan. 2018.
[11] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, Oct. 2017.

[12] M. A. Tawhid and K. B. Dsouza, "Hybrid binary bat enhanced particle swarm optimization algorithm for solving feature selection problems," *Appl. Comput. Informat.*, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S221083271830005X, doi: 10.1016/j.aci.2018.04.001.

[13] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, Feb. 2019.

[14] S. Arora and S. Singh, "Node localization in wireless sensor networks using butterfly optimization algorithm," *Arabian J. Sci. Eng.*, vol. 42, no. 8, pp. 3325–3335, Aug. 2017.

[15] S. Arora and P. Anand, "Learning automata-based butterfly optimization algorithm for engineering design problems," *Int. J. Comput. Mater. Sci. Eng.*, vol. 7, no. 4, Dec. 2018, Art. no. 1850021.

[16] L. El Hajjami, E. M. Mellouli, and M. Berrada, "Optimal PID control of an autonomous vehicle using butterfly optimization algorithm BOA," in *Proc. 4th Int. Conf. Big Data Internet Things*, Oct. 2019, pp. 1–5.

[17] S. Arora and S. Singh, "An effective hybrid butterfly optimization algorithm with artificial bee colony for numerical optimization," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 4, no. 4, p. 14, 2017.

[18] S. Arora and S. Singh, "A hybrid optimisation algorithm based on butterfly optimisation algorithm and differential evolution," *Int. J. Swarm Intell.*, vol. 3, nos. 2–3, p. 152, 2017.

[19] S. Arora and S. Singh, "An improved butterfly optimization algorithm with chaos," *J. Intell. Fuzzy Syst.*, vol. 32, no. 1, pp. 1079–1088, Jan. 2017.

[20] 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.

[21] S. Mirjalili and A. Lewis, "S-shaped versus V-shaped transfer functions for binary particle swarm optimization," *Swarm Evol. Comput.*, vol. 9, pp. 1–14, Apr. 2013.

[22] S. Taghian, M. H. Nadimi-Shahraki, and H. Zamani, "Comparative analysis of transfer function-based binary Metaheuristic algorithms for feature selection," in *Proc. Int. Conf. Artif. Intell. Data Process. (IDAP)*, Sep. 2018, pp. 1–6.

[23] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. IEEE Int. Conf. Syst., Man, Cybern., Comput. Cybern. Simul.*, vol. 5, Oct. 1997, pp. 4104–4108.

[24] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "BGSA: Binary gravitational search algorithm," *Natural Comput.*, vol. 9, no. 3, pp. 727–745, Sep. 2010.

[25] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[26] S. K. Nayak, P. K. Rout, A. K. Jagadev, and T. Swarnkar, "Elitism based multi-objective differential evolution for feature selection: A filter approach with an efficient redundancy measure," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 2, pp. 174–187, Feb. 2020.

[27] 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.

[28] M. M. Gouvea and A. F. R. Araujo, "Population dynamics model for gene frequency prediction in evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 1603–1610.

[29] J. Franklin, "The elements of statistical learning: Data mining, inference and prediction," *Math. Intelligencer*, vol. 27, no. 2, pp. 83–85, Mar. 2005.

[30] A. Asuncion and D. J. Newman, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, 2007. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

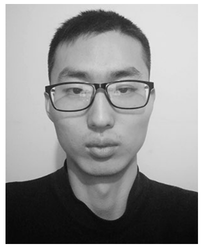
[31] M. Ghaemi and M.-R. Feizi-Derakhshi, "Feature selection using forest optimization algorithm," *Pattern Recognit.*, vol. 60, pp. 121–129, Dec. 2016.

[32] E. Hancer, B. Xue, M. Zhang, D. Karaboga, and B. Akay, "Pareto front feature selection based on artificial bee colony optimization," *Inf. Sci.*, vol. 422, pp. 462–479, Jan. 2018.

[33] Y. Zhang, A. Yang, C. Xiong, T. Wang, and Z. Zhang, "Feature selection using data envelopment analysis," *Knowl.-Based Syst.*, vol. 64, pp. 70–80, Jul. 2014.

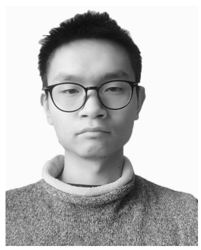
[34] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms," *Appl. Soft Comput.*, vol. 18, pp. 261–276, May 2014.

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

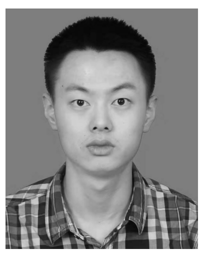


BO ZHANG is currently pursuing the B.S. degree in software engineering with Jilin University. His current research interests include intelligence optimization and feature selection.

XINKAI YANG is currently pursuing the B.S. degree in software engineering with Jilin University. His current research interests include feature selection and constraint reasoning, intelligence optimization, and feature selection.



BIAO HU is currently pursuing the B.S. degree in software engineering with Jilin University. His current research interests include feature selection and evolutionary computation.



ZHAOGENG LIU is currently pursuing the master's degree with Jilin University. His research interests include feature selection and evolutionary computation. He was a recipient of a number of awards, including the National Scholarship, in 2019, the National Scholarship, in 2018, and so forth.

ZHANSHAN LI is currently a Professor of computer science with Jilin University. His main research interests include constraint reasoning and machine learning.