Evolutionary Intelligence (2023) 16:1653–1666
https://doi.org/10.1007/s12065-023-00819-1

SPECIAL ISSUE

An improved binary particle swarm optimization combing V-shaped and U-shaped transfer function

Yuxiang Chen[1,2] · Jianhua Liu[1,2] · Jian Zhu[1,2] · Zihang Wang[1,2]

Received: 29 September 2022 / Revised: 21 November 2022 / Accepted: 31 December 2022 / Published online: 21 January 2023

Abstract
Feature selection aims to find a best feature subset from all feature sets of a given dataset, which represents the whole feature space to reduce redundancy and improve classification accuracy. The evolutionary computation algorithm is often applied to feature selection, but there exists low efficiency in the search process. With the increase of the number of features, solving the feature selection problem become more and more difficult. Existing evolutionary algorithms have many defects, such as slow convergence speed, low convergence accuracy and easy to fall into local optimum. Therefore, the research of more effective evolutionary algorithms has important theoretical significance and application value. Binary Particle Swarm Optimization (BPSO) is a kind of evolutionary computation algorithm and has a good performance in feature selection problems. It uses transfer function to convert the continuous search space to the binary one. Transfer function plays an important role in BPSO. So this paper proposes an improved BPSO by combining V-shaped and U-shaped transfer function, and introduces a new learning strategy and a local search strategy based on adaptive mutation. The improved BPSO enhances its optimization ability in feature selection problem. The experimental results show that the improved BPSO has better dimension reduction ability and classification performance than other algorithms.

Keywords Feature selection · Classification · Particle swarm optimization (PSO) · Transfer function · Evolutionary computation

1 Introduction

In real-world applications, there are a variety of data, such as medical data, web documents and music data. These data often contain many features, 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. Thus, feature selection is very important.

Feature selection is a preprocessing step in machine learning. It is a process of searching an optimal subset in the original feature space according to certain evaluation criteria from a dataset, so that making machine learning algorithms to perform faster and more effective. And feature selection can make machine learning methods to perform better while saving costs [1].

The aim of feature selection is to select some representative and important features [2], and the commonly used methods can be classified into 3 categories according to the criterion used: filter method, wrapper method, and embedded method [3]. Filter method evaluates the quality of features in an unsupervised way, while wrapper method evaluates the quality of them in a supervised way. Filter method generally uses evaluation criteria to enhance the relevance between features and classes and reduce the redundancy of features. Wrapper method takes feature selection algorithm as a part of machine learning algorithm, and uses classification performance as the evaluation criteria. Embedded combines the process of feature selection with the training process of machine learning. In the training process, feature

✉ Yuxiang Chen
541893237@qq.com

✉ Jianhua Liu
jhliu@fjnu.edu.cn

1 School of Computer Science and Mathematics, Fujian University of Technology, Fuzhou, Fujian, China

2 Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350118, Fujian, China

Report, feature

selection is carried out automatically. Moreover, the embedded method is actually the special cases of the wrapper methods since the feature selections are regarded as a part of training phase in machine learning [4]. In general, filter method is more time-saving and widely used than wrapper method, while wrapper method has better performance than filter method.

Heuristic search is a way to complete and random search for generating feature subsets. Several metaheuristics including Particle Swarm Optimization (PSO) [5], Manta Ray Foraging Optimization (MRF) [6], Whale Optimization Algorithm (WOA) [7], Ant Lion Optimization (ALO) [8], Gray Wolf Optimizer (GWO) [9] and Remora Optimization Algorithm (ROA) [10] may approved notable capabilities in tackling FS problems. Moreover, FS has been utilized to resolve many classification problems belong to diverse fields like data mining [11], pattern recognition [12], power dispatch [13] and others where FS can be utilized.

In recent years, PSO-based wrappers have been studied more than filters. PSO are extensively employed in the wrappers, however, the algorithm suffers from some disadvantages when solving the complex and high-dimensional objective functions. PSO may easily get trapped in local optima and show slow convergence rate [14]. Hence, many literatures have proposed different BPSO variants to improve the algorithm. They improved the performance of algorithms in different ways such as selecting parameters, combining with other search techniques, and so on. Kennedy and Eberhart [15] introduced a sigmoid function for binary PSO (SBPSO). This function is the base of S-shaped transfer functions. In the BPSO, if the velocity is positive, the next value is more likely to be 1. By contrast, when the velocity is negative, the next value is more likely to be 0. If the velocity tends to be zero, the sigmoid function will be equal to 0.5; therefore, the next position may be one or zero. This will lead to strong randomness of BPSO algorithm. Although it has global exploration ability, it will not converge to the global optimal particle. As the number of iterations increases, its randomness will gradually increase, which will lead to its lack of local search ability.

Since the appearance of the first version of Binary PSO in 1997, many researchers have tried to enhance its ability in solving various discrete optimization problems by introducing several ideas. For instance, Li et al. [16] propose an improved sticky binary PSO (ISBPSO) algorithm for FS. ISBPSO adopts three new mechanisms based on a recently proposed binary PSO variant, sticky binary particle swarm optimization (SBPSO) to improve the evolutionary performance. Song et al. [17] propose a novel feature selection algorithm based on bare bones PSO (BBPSO) with mutual information. They introduce a swarm initialization strategy based on label correlation is developed, making full use of the correlation between features and class labels to accelerate the convergence of swarm. Beheshti et al. [18] propose a binary hybrid topology particle swarm optimization (BHTPSO) to solve the optimization problems in the binary search spaces. The proposed algorithms have significantly lower error than the others. Shen et al. [19] proposed a modified BPSO for selecting variables in MLR and PLS modeling. In addition, Wang et al. [20] proposed a novel probability-based BPSO algorithm and evaluated its performance in solving the multidimensional knapsack problem. Chuang et al. [21] introduced an enhanced BPSO via employing the catfish effect and applied it for feature selection.

The transfer function plays an important part in the process of convert the continuous search space to the binary search space. In order to solve the problems caused by S-shaped transfer functions, various types of transfer functions have been proposed in recent years such as linear [20], V-shaped [22], U-shaped [23], Z-shaped [24], X-shaped [25]. On the other hand, a time varying (TV) transfer function was proposed to enhances global exploration and local exploitation in BPSO [26]. But the S-shaped and V-shaped functions are the bases of the TV transfer function so it is facing the same problems as theirs.

In S-shaped transfer function, a big value in the positive direction of velocity increases the probability of one for the next position and it is zero for the negative direction. If the velocity tends to be zero, the transfer function will be equal to 0.5, this lets the swarm diverge in the last iterations. The Z-shaped transfer function is also faced with the drawbacks of S-shaped BPSO. Because of the high flip probability, the U-shaped transfer function's convergence speed is very slow. In V-shaped transfer function, a big value in the negative and positive directions shows that a great movement is needed to achieve the best solution. When the velocity tends to be zero, the next position will be the current position. Therefore, if the current position is a local optimum, the next position will be the local optimum. In other words, if the best position found by swarm is the local optimum, the population will fall into the local optimum. Besides, X-shaped transfer function is time-consuming. It has poor local searching ability, which make it inefficient in solving high-dimensional problems.

As mentioned above, although many transfer functions have been proposed in the literature, they also have certain defects. It motives us to improve the conventional BPSO so that making it more suitable for the feature selection.

The main contributions of this paper are summarized as follows:

- We propose a new transfer function combining V-shaped and U-shaped transfer functions and apply in BPSO (CVUBPSO) to solve the formulated feature selection problem. Then, we introduced a new learning strategy

and a local search strategy based on adaptive mutation to improve its performance.
- We conducted a series of experiments on 10 classic datasets obtained from UCI repository to evaluate the performance of the proposed algorithm.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 introduce the CVUBPSO algorithm. Section 4 discusses the experiment results and Sect. 5makes a summary of the studies in this paper.

2 Related work

This section provides the basic concepts of BPSO and transfer function, which are the basic components of the proposed approach, followed by an overview of related work on FS.

2.1 Conventional BPSO

In PSO, each particle i in dimension d is associated with two vectors, that is, a position vector $X_i = [x_i^1, x_i^2,\ldots, x_{i,}^D]$ and a velocity vector $V_i = [v_i^1, v_i^2,\ldots, v_i^D]$, where D represents the dimension of a problem under study. During the evolutionary process, each particle adjusts its flight trajectory based on two vectors, called personal historical best position $pbest_i^d = [pbest_i^1, 1, pbest_i^2,\ldots, pbest_i^D]$ and the current global best position $gbest$, respectively. The update rules of V_i and X_i are defined as (1) and (2), respectively.

$$v_i^d(t+1) = \omega * v_i^d(t) + c_1 * r_1 * \left(pbest_i^d - x_i^d(t)\right) + c_2 * r_2 * \left(gbest - x_i^d(t)\right) \tag{1}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{2}$$

where w represents an inertia weight determining how much the previous velocity is preserved; c_1 and c_2 are two acceleration coefficients deciding learning weights for $pbest_i^d$ and $gbest$, respectively; r_1 and r_2 are two random numbers uniformly distributed in the interval [0, 1]; and x_i^d and v_i^d represent the dth dimension values of X_i and V_i, respectively.

The following transfer function is employed in PSO to convert BPSO:

$$Tf\left(v_i^d(t+1)\right) = \frac{1}{1 + e^{-v_i^d(t+1)}} \tag{3}$$

where $v_i^d(t+1)$ is the next velocity of the i particle in the dth dimension, $|v_i^d(t+1)|$ can't exceed the maximum velocity v_{max}, and v_{max} is set to a constant. The sigmoid function generates a value between 0 and 1 according to the current velocity then updates the next position, $x_i^d(t+1)$ is calculated as follows:

$$x_i^d(t+1) = \begin{cases} 1 \ if \ rand < Tf\left(v_i^d(t+1)\right) \\ 0 \ if \ rand \geq Tf\left(v_i^d(t+1)\right) \end{cases} \tag{4}$$

where rand is a random number in U ~ (0,1).

The next position in BPSO is not generated based on the current position. In the algorithm, if the velocity is big in positive position, the next position will be one with more probability. Also, the next position will be zero in the negative direction.

2.2 Transfer function

To overcome the drawbacks of S-shaped transfer functions, a V-shaped transfer function is proposed by Rostami et al. [27] as follows:

$$Tf\left(v_i^d(t+1)\right) = \left|\tanh\left(\alpha \cdot v_i^d(t+1)\right)\right|$$

The shape of the V-shaped transfer function is shown in Fig. 1. The larger the velocity, the bigger the probability of changing the current position and vice versa. The function is classified in the V-shaped transfer functions. V-shaped transfer functions suffer from some disadvantages. If the current position is in the local optimal optimum, the position of the next generation may also be in the local optimal. This will eventually lead to the algorithm falling into the local optimal. To solve this problem, Nezamabadi-pour et al. [27] suggested a new parameter for V-shaped transfer functions to escape from the local optimum as follows:

$$Tf(v_i^d(t+1) = A + (1-A) * \left|\tanh\left(\alpha v_i^d(t+1)\right)\right|$$

where A is a parameter to avoid the stagnation during the algorithm running. A is defined as follows:

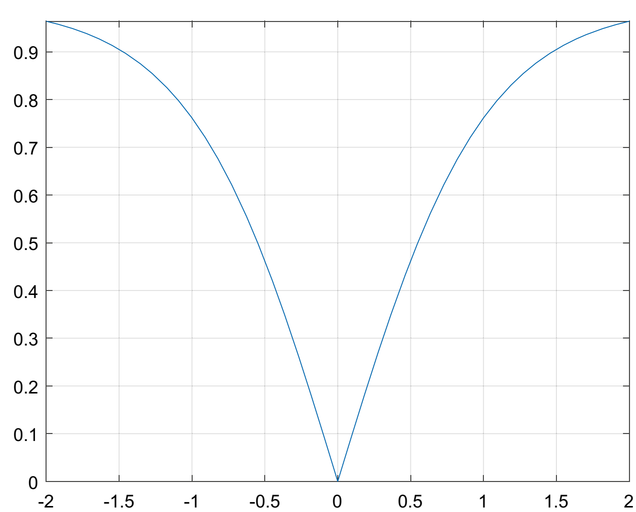

Fig. 1 The shape of V-shaped transfer function

$$A = k\left(1 - e^{-\frac{F}{T}}\right)$$

k is a constant. T and F represent the dimensions of particle and the number of times to stop updating of global optimal solution. If the stagnation does not happen F will be zero. Otherwise F will be increased by one.

Several U-shaped family of transfer functions were proposed by Mirjalili et al. [23] to convert the continuous PSO to the binary PSO. The rule of V-shaped functions is applied to generate the next binary position. The transfer function faces the drawbacks of V-shaped transfer functions and also, the results are highly dependent on the slope and the width of U-shaped transfer function. Its shape is shown in Fig. 2.

$$Tf(v_i^d(t+1)) = \alpha\left|v_i^d(t+1)^{\beta}\right|$$

where α and β are two control parameters as the slope and the width of U-shaped transfer function, respectively.

Guo et al. [24] proposed a new Z-shaped transfer function in Fig. 3 and conduct experiments on three sets' data show that the newly proposed Z-shaped transfer function improves the convergence speed and convergence precision of BPSO algorithm. The transfer function is defined as follows:

$$\mathrm{T}\left(x_i^k(t)\right) = \sqrt{1 - a^{x_i^k(t)}}$$

where $x_i^k(t)$ represents the velocity of particle i at iteration t in k dimension and a is a positive integer.

Beheshti [25] proposed new X-shaped transfer function in Fig. 4. In X-shaped BPSO (XBPSO), the new position is generated as follows:

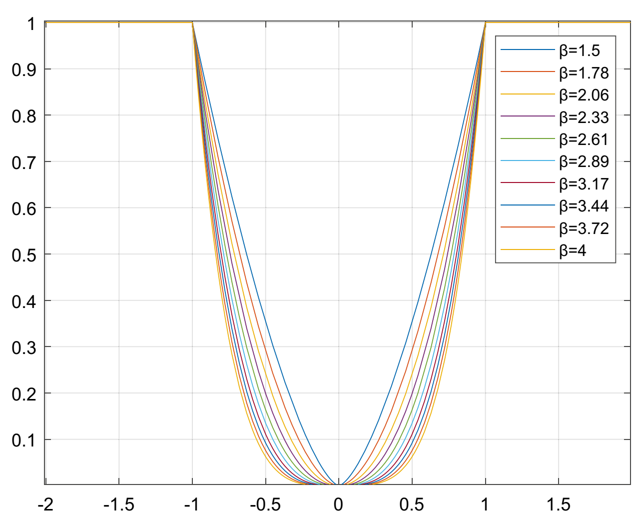

Fig. 2 The shape of U-shaped transfer function

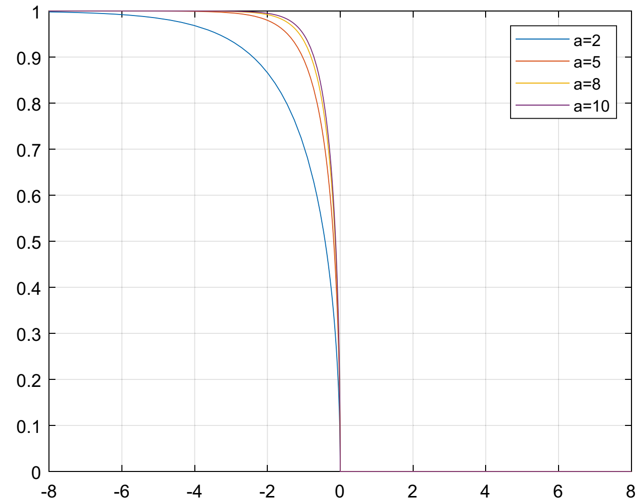

Fig. 3 The shape of Z-shaped transfer function

$$S_1\left(v_i^d(t+1)\right) = \frac{-v_i^d(t+1)}{1 + \left|-v_i^d(t+1)\right| * 0.5} + 0.5$$

$$y_i^d = \begin{cases} 1 \text{ if } \mathrm{rand}_1 > S_1\left(v_i^d(t+1)\right) \\ 0 \text{ if } \mathrm{rand}_1 \le S_1\left(v_i^d(t+1)\right) \end{cases}$$

$$S_2\left(v_i^d(t+1)\right) = \frac{v_i^d(t+1) - 1}{1 + \left|v_i^d(t+1) - 1\right| * 0.5} + 0.5$$

$$z_i^d = \begin{cases} 1 \text{ if } \mathrm{rand}_1 > S_2\left(v_i^d(t+1)\right) \\ 0 \text{ if } \mathrm{rand}_1 \le S_2\left(v_i^d(t+1)\right) \end{cases}$$

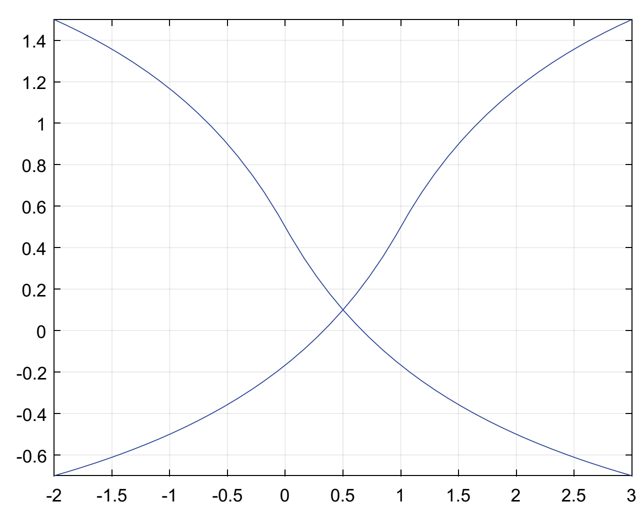

Fig. 4 The shape of X-shaped transfer function

Two new positions y_i and z_i are created by two functions. Therefore, y_i and z_i are two different solutions generated based on different probabilities to improve the ability of algorithm for finding better solution. The best position is selected as follows:

$$P_i(t+1) = \begin{cases} y_i \text{ if } f(y_i) \text{ is better than} f(z_i) \\ z_i \text{ if } f(z_i) \text{ is better than } f(y_i) \end{cases}$$

where f(.) is the fitness function.

A time-varying transfer function was introduced by Islam et al. [26]. The algorithm, namely TV-BPSO, was evaluated by combinatorial problems for low and high dimension knapsack problems. Its shape is shown in Fig. 5. The method applied the following transfer function and the next position is created based on S-shaped transfer functions:

$$S\left(v_i^d(t+1), \phi\right) = \frac{1}{1 + e^{\frac{-v_i^d(t+1)}{\phi}}}$$

$$\phi = \phi_{max} - iter\left(\frac{\phi_{max} - \phi_{min}}{\text{maxiter}}\right)$$

where φ changes the slope of sigmoid function. ϕ_{max} and ϕ_{min} are the maximum and minimum bounds of φ. t is the current iteration and max iteration presents the maximum number of iterations. Although TV-BPSO has improved the balance between exploration and exploitation, it still faces the shortcomings of employing the sigmoid function as the base of transfer function.

To sum up, previous literature has made different improvements to the transfer function but insufficiency still exists. A good transfer function should create balance between exploration and exploitation to achieve a proper solution so as to enhance the performance of the meta heuristic algorithm.

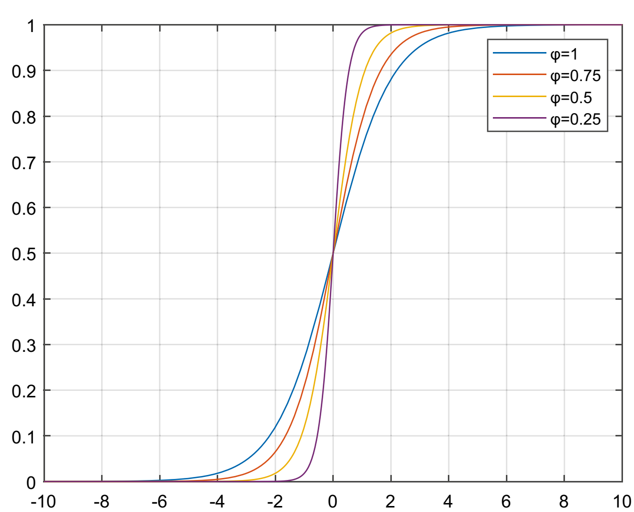

Fig. 5 The shape of TV-shaped transfer function

2.3 Formulation of feature selection problem

The feature selection is a multi-objective optimization problem. It has two objectives: minimizing the number of selected features and maximizing the classification accuracy. These two objectives can be represented in a fitness function as follows:

$$f_{FS} = a \cdot E + b \cdot \frac{d}{D} \tag{5}$$

where E is the classification error. d and D are the numbers of selected features and the numbers of dataset features, respectively. d/D is feature reduction rate. a and b are the weight of classification error rate and feature reduction rate respectively. a is in interval of [0, 1] and $b = 1 - a$.

In this paper, the k-nearest neighbor (KNN) method is used for feature selection to evaluate the classification accuracy. The KNN is a non-parametric classification method, which is simple but effective in many cases. This method classifies the data t to be classified, retrieves its k nearest neighbors, and forms the neighborhood of t. The class that gets the most votes in the neighborhood is used to determine the class of t [28].

3 The proposed CVUBPSO algorithm

The proposed algorithm called CVUBPSO, is described in this section. First, the new transfer function combining V-shaped and U-shaped transfer function is introduced in Sect. 3.1 Next, the new learning strategy to enhance the diversity and information transmission ability of swarm is clarified in Sect. 3.2. Finally, the injection of the adaptive mutation operators into CVUBPSO is describe in Sect. 3.3.

3.1 A novel transfer function

From the previous research, it can be seen that the search ability of most transfer functions is still insufficient. Some transfer functions work well when dealing with low dimensional data but they do not work well when dealing with high dimensional data, while others do the opposite. In order to solve this problem, we combine the V-shaped transfer function and the U-shaped transfer function to construct a new type of transfer function.

The proposed transfer function combines the advantages of V-shaped transfer function and U-shaped transfer function. Flipping the bits more frequently when using the U-shaped transfer function allows the BPSO to avoid the local solutions in the feature selection problems. However, U-shaped transfer function has poor performance when dealing with low dimensional problems. Although V-shaped transfer function is easy to fall into local optimization, its search ability in low dimensional space is stronger than U-shaped transfer function. Thus, we combine it with the V-shaped transfer function. It upgrades the performance of algorithm to balance exploration and exploitation for obtaining better results. Figure 6 shows the shape of proposed transfer function with other U-shaped and V-shaped transfer functions. As the figure shows, the proposed transfer function has a larger opening in the center than other V-shaped or U-shaped transfer function. When the velocity value is close to 0, particles can converge faster. This improves the ability of the algorithm to deal with high-dimensional problems. When the velocity value is too large or small, the value of the function is gradually close to 1. That conducive to prevent particles from moving closer quickly. That can prevent algorithm from falling into local optimum, which is very effective in dealing with low dimensional problems. The transfer functions are defined as follows:

$$Tf\left(v_i^d(t+1)\right) = \frac{2}{\pi}\arctan\left(x^2\right) \tag{6}$$

where $v_i^d(t)$ and $v_i^d(t+1)$ are the current velocity and the next velocity of the ith particle in the dth dimension, respectively. $i=1,2,\ldots,n$ (n is the number of particles) and $d=1,2,\ldots,D$ (D is the number of dimensions).

The next positions of x_i^d is computed as follows:

$$x_i^d(t+1) = \begin{cases} \neg x_i^d(t) \ if \ rand < Tf\left(v_i^d(t)\right) \\ x_i^d(t) \ if \ rand \geq Tf\left(v_i^d(t)\right) \end{cases} \tag{7}$$

where $rand$ is a random number in U ~ (0,1).

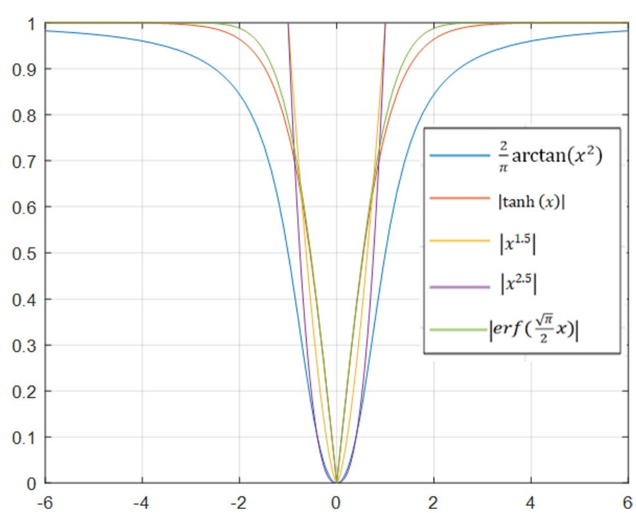

Fig. 6 The shape of proposed transfer function with some other U-shaped and V-shaped transfer functions

The first advantage of this transfer function is it has a larger search space. The second advantage of it is can deal with both low dimensional and high dimensional problems.

3.2 A new learning strategy

In feature selection problem, feature subset with the same size but different features usually have different classification errors. Simply select the solution with better classification accuracy may neglect the diversity of solutions. Although the proposed transfer function greatly enhances the search ability of the algorithm, particles are still prone to lose some important information in the learning process, resulting in the algorithm falling into local optimization. Hence, a new PSO update strategy considering both information transmission and diversity of solutions is designed to guide the search process. To solve this problem, we propose a new learning strategy. The new velocity updating equations of are as follows:

$$\begin{aligned} v_i^d(t+1) = {} & \omega * v_i^d(t) + c1 * r1 * \left(pbest_i^d - x_i^d(t)\right) \\ & + c2 * r2 * \left(gbest - x_i^d(t)\right) \\ & + c3 * r3 * \left(mnbest_i^d - x_i^d(t)\right) \end{aligned} \tag{8}$$

$$mnbest_i^d = \frac{nbest_{i,1}^d + nbest_{i,2}^d}{2} \tag{9}$$

where $nbest_{i,1}^d$ and $nbest_{i,2}^d$ are the two neighbors of the current particle, mnbest_i^d is the mean value of them.

It is very important for a particle to select proper neighbors since much useful knowledge extracted from the exemplars can provide some constructive guidance for the particle. We use ring topology for the population. In particular, the particles are sorted randomly, and then a particle selects two particles adjacent to it as the neighbors of it. That can ensure the diversity of the population. The mean value of the personal best position of the particle's neighbors is used as its learning example, so that the information between particles in the iteration process can be fully communicated.

3.3 Local search strategy based on adaptive mutation

In the new learning strategy, the communication and diversity between particles are enhanced but this strategy only enhances the global search ability at the beginning of iteration. We need to use another strategy to enhance the local search ability of the algorithm in the later stage of iteration. In the CVUBPSO, each particle is updated by its own personal best position and global best position. If the global best position falls into local optimum, the whole algorithm may fall into local optimum. Thus, to further enhance the exploitation ability of conventional CVUBPSO, a local search operator based on adaptive mutation mechanism for *gbest* is proposed. In this method, the mutation probability of each dimension of *gbest* changes with the average value of all particle's personal historical best position. This can make the global best particles change according to the evolution of the population, and effectively prevent the population from converging prematurely and falling into local optimization. Thus, the search scope was expanded and the local search ability was enhanced. Accordingly, the proposed mutation probability P_d is described as follows:

$$P_d = 1 - \left(\frac{1}{N} \sum_{i=1}^{N} pbest_i^d - 0.5 \right)^2 * 4 \tag{10}$$

where N is the population size and $pbest_i^d$ is the ith particle's personal historical best position in the dth dimension. In our strategy, if a randomly generated number rand is less than P_d, then the corresponding dimension of solution should be mutated to its inverse value. Figure 7 shows the shape of this function. As can be seen from the figure, the closer the average value of $pbest^d$ is to 0.5, the more probably of the position will change. The main steps of the local search strategy based on adaptive mutation is shown in Algorithm 1. Accordingly, the main steps of CVUBPSO are shown in Fig. 8.

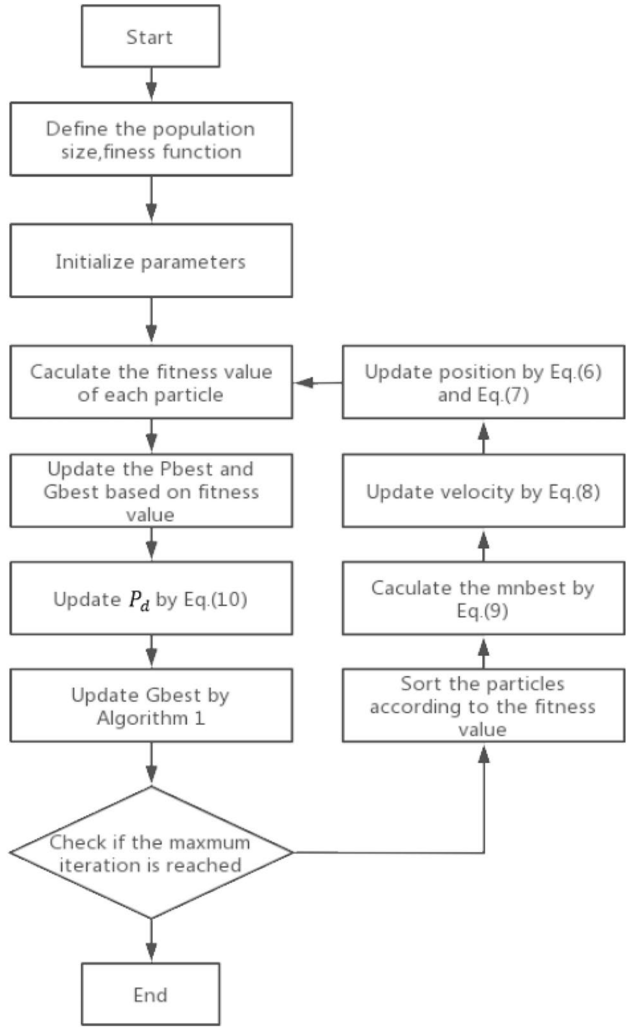

Fig. 8 Flowchart of the CVUBPSO algorithm

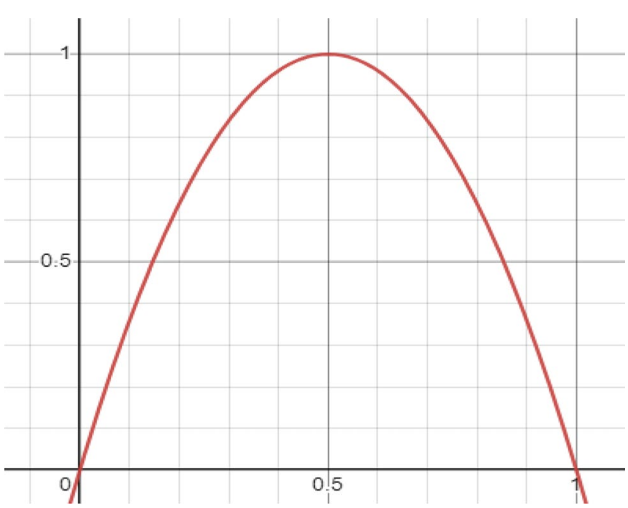

Fig. 7 The shape of the mutation probability

Algorithm 1 Local search strategy based on adaptive mutation

```
Calculate the value of P_d by using Eq. (10);
X^d_gbest2 = X^d_gbest;
for d = 1 to D do
    if rand < P_d then
    X^d_gbest2 = ~X^d_gbest2;
    end
end
f = evaluate X_gbest2 by Eq. (5);
if f < f_gbest then
    f_gbest = f;
    X_gbest = X_gbest2;
end
```

4 Experimental result and analysis

In order to prove the effectiveness of the improved strategy and the superiority of the improved algorithm, two groups of comparative experiments are set up. The first group of experiments compares the results of the proposed algorithm with other algorithms. The second group of comparative experiments verify effectiveness of the proposed transfer function, the new learning strategy and local search strategy based on adaptive mutation in this paper respectively.

4.1 Details of datasets and experiment setups

The details of datasets used in the evaluations of different algorithms are introduced.

4.1.1 Details of datasets

The proposed approaches were tested on 10 feature selection datasets obtained from the UCI [29] data repository. Table 1 shows characteristics of these datasets.

4.1.2 Experiment setups

In the comparison experiment, the genetic algorithm (GA) [30], binary ant lion optimizer (BALO) [8], binary cuckoo search (BCS) [31] and V-shaped transfer function (VBPSO) [27] are introduced as the comparison algorithms. In addition, the maximum number of iterations for each algorithm is set as 100, the population size is 20, and the dimension of solution is equal to the feature number of each dataset. Each algorithm executes feature selection on these data sets and runs independently for 30 times. Moreover, in each dataset, we adopt the hold-out strategy. We used 80% of the data for training, and the rest ones are used for testing, which is a common way adopted by feature selection. All the parameter of the algorithms are shown in Table 2.

Table 1 Details of datasets

Dataset	Features	Samples
Breastcancer	9	699
HeartEW	13	270
IonosphereEW	34	351
SonarEW	60	208
SpectEW	22	267
WaveformEW	40	5000
penglungEW	325	73
Libras Movement	90	360
Hill Valley	100	1212
Urban-Land-Cover	147	675

Table 2 Parameter setups of different algorithms

Algorithm	Value of key parameters
GA	Mutation rate = 0.05, Crossover rate = 0.5
BCS	Discovery probability = 0.25, α=1
BALO	$r_0 \in [0.9\ 0]$
VBPSO	c1 = 1.5, c2 = 2.0
CVUBPSO	c1 = 1.05, c2 = 1.05, c3 = 1.05

4.2 Feature selection results

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

4.2.1 Fitness value evaluations in different algorithms

In this section, the fitness function values obtained by different algorithms are presented to show the performances of these approaches directly. Table 3 shows the numerical statistics results in terms of standard deviation (SD) and average value of different algorithms for each datasets, and the best values obtained by a certain algorithm are highlighted in bold font for a clear presentation. It can be seen from the tables that the proposed CVUBPSO algorithm achieves the best average fitness function values on majority of datasets, which means it has better performance than other comparison algorithms.

In addition, the convergence rates of different algorithms during the processes of solving the fitness functions are shown in Fig. 9. Note that these curves are the mean fitness value. As can be seen, the convergence accuracy and convergence speed of the CVUBPSO are superior to other algorithms.

4.2.2 Feature selection accuracies in different algorithms

The feature selection accuracies obtained by different algorithm are presented in Table 4. Similarly, the numerical statistics results of different algorithms for each dataset are presented in these tables. On the whole, all approaches achieve the same results on Breastcancer dataset and this may be that these dataset is with the lower solution dimension, which makes it easy to be solved. As can be seen, CVUBPSO algorithm achieves the best average accuracies of feature selection results on majority of datasets. Thus, CVUBPSO 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

Table 3 Fitness values obtained by different algorithms

Dataset	Algorithm	Mean	Std
Breastcancer	GA	0.0037	2.23E−03
	BCS	**0.0269**	3.47E−18
	BALO	**0.0269**	3.47E−18
	VBPSO	**0.0269**	3.47E−18
	CVUBPSO	**0.0269**	3.47E−18
HeartEW	GA	0.0789	1.56E−02
	BCS	0.0415	9.76E−04
	BALO	0.0416	9.23E−04
	VBPSO	0.0449	1.05E−02
	CVUBPSO	**0.0411**	7.69E−04
IonosphereEW	GA	0.1056	1.13E−02
	BCS	0.0726	7.38E−03
	BALO	0.0547	1.10E−02
	VBPSO	0.0393	1.15E−02
	CVUBPSO	**0.0316**	1.30E−02
SonarEW	GA	0.1061	1.45E−02
	BCS	0.0748	1.28E−02
	BALO	0.0695	1.52E−02
	VBPSO	0.0450	1.55E−02
	CVUBPSO	**0.0302**	1.67E−02
SpectEW	GA	0.1888	2.05E−02
	BCS	0.1413	8.30E−03
	BALO	0.1218	1.44E−02
	VBPSO	**0.1287**	2.32E−02
	CVUBPSO	0.1106	1.20E−02
WaveformEW	GA	0.2079	6.84E−03
	BCS	0.1777	3.81E−03
	BALO	0.1591	4.21E−03
	VBPSO	0.1542	5.35E−03
	CVUBPSO	**0.1462**	3.52E−03
PenglungEW	GA	0.0118	2.11E−02
	BCS	0.0043	7.69E−05
	BALO	0.0036	1.75E−04
	VBPSO	0.0031	1.80E−04
	CVUBPSO	**0.0018**	1.08E−04
Libras movement	GA	0.1549	4.20E−03
	BCS	0.1297	7.35E−03
	BALO	0.1123	9.66E−03
	VBPSO	0.1067	6.90E−03
	CVUBPSO	**0.0811**	1.10E−02
Hill Valley	GA	0.4106	1.71E−02
	BCS	0.3919	6.89E−03
	BALO	0.3538	7.11E−03
	VBPSO	0.3577	1.16E−02
	CVUBPSO	**0.3232**	1.05E−02
Urban-Land-Cover	GA	0.1290	9.87E−03
	BCS	0.1095	4.52E−03
	BALO	0.0945	8.62E−03
	VBPSO	0.0895	7.62E−03
	CVUBPSO	**0.0752**	1.14E−02

and exploitation abilities, so that enhancing the performance of the algorithm.

4.2.3 Number of selected features in different algorithms

Table 5 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 CVUBPSO obtains the best average number of selected features over half of the datasets (8 of 10), which can be regarded as the best results in the tests compared to other algorithms. The classification accuracy and the number of selected features need to be trade-off. Even if the number of selected features is large, the classification accuracy may not be low. Thus, we can conclude that our algorithm has certain advantages in feature selection compared with other algorithms.

4.3 Effectiveness of the improved strategy

In this section, we conduct tests to evaluate the effectiveness of the improved factors in CVUBPSO. The results are discussed in detail as follows:

4.3.1 A novel transfer function

In this section, the proposed transfer function is compared with other transfer functions. Note that CVUBPSO for comparison there does not use any improved strategy. These transfer functions are applied in BPSO to evaluate the performance (Table 6).

It can be seen from Table 7 that the fitness function value obtained by the proposed transfer function are better than others. It shows that the proposed transfer function can make the algorithm achieve a good balance between exploration and exploitation.

4.3.2 A new learning strategy

We investigate the effectiveness of the new learning strategy by comparing it with the original CVUBPSO. Table 8 reports that for the 10 datasets, the performance of the algorithm with the new learning strategy is better than that without it. For each dataset, the algorithm with the proposed new learning strategy obtains the best classification accuracies. This proves that the strategy is effective in improving the performance of the algorithm.

4.3.3 Local search strategy based on adaptive mutation

Table 9 shows that the accuracies of fitness function values obtained by the local search strategy based on adaptive

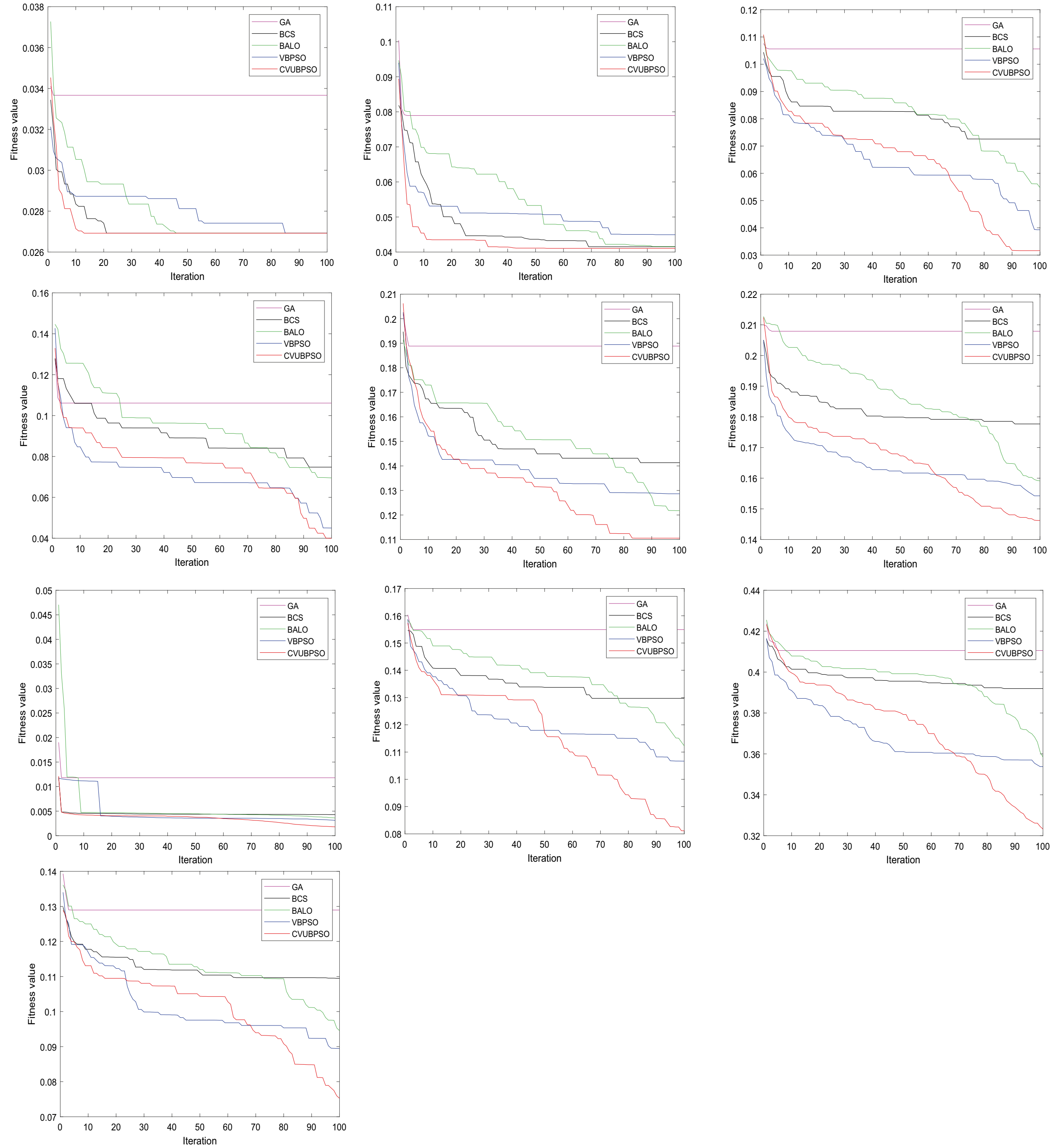

Fig. 9 Mean fitness value comparison graph of 10 dataset

mutation (CVUBPSO-LSAM) performs better than that without the strategy, especially on low-dimensional datasets. This is because that the proposed local search strategy based on adaptive mutation improving the exploration ability. This strategy can enhance the local search ability of the algorithm which provide a better exploitation for finding better global best solution. Then, it makes the algorithm achieve higher fitness value.

Above all, the proposed transfer function and the two strategies are complementary and effectively improve the performance of the BPSO. According to the experimental result, a conclusion can be derived that CVUBPSO not

Table 4 Classification accuracies obtained by different algorithms

Dataset	Algorithm	Mean	Std
Breastcancer	GA	0.9719	2.16E−03
	BCS	**0.9784**	1.11E−16
	BALO	**0.9784**	1.11E−16
	VBPSO	**0.9784**	1.11E−16
	CVUBPSO	**0.9784**	1.11E−16
HeartEW	GA	0.9259	1.66E−02
	BCS	0.9630	1.11E−16
	BALO	0.9630	1.11E−16
	VBPSO	0.9593	1.11E−02
	CVUBPSO	**0.9630**	1.07E−02
IonosphereEW	GA	0.8971	1.14E−02
	BCS	0.9300	7.69E−03
	BALO	0.9471	1.12E−02
	VBPSO	0.9629	1.14E−02
	CVUBPSO	**0.9700**	1.29E−02
SonarEW	GA	0.8971	1.46E−02
	BCS	0.9293	1.31E−02
	BALO	0.9341	1.56E−02
	VBPSO	0.9585	1.56E−02
	CVUBPSO	**0.9732**	1.83E−02
SpectEW	GA	0.8132	2.14E−02
	BCS	0.8623	8.65E−03
	BALO	0.8811	1.47E−02
	VBPSO	0.8736	2.39E−02
	CVUBPSO	**0.8925**	1.25E−02
WaveformEW	GA	0.7954	6.56E−03
	BCS	0.8261	3.86E−03
	BALO	0.8448	4.79E−03
	VBPSO	0.8497	5.08E−03
	CVUBPSO	**0.8579**	3.67E−03
PenglungEW	GA	0.9929	2.14E−02
	BCS	**1.0000**	0.00E+00
	BALO	**1.0000**	0.00E+00
	VBPSO	**1.0000**	0.00E+00
	CVUBPSO	**1.0000**	0.00E+00
Libras movement	GA	0.8486	4.17E−03
	BCS	0.8736	7.48E−03
	BALO	0.8903	9.72E−03
	VBPSO	0.8958	6.94E−03
	CVUBPSO	**0.9208**	1.11E+02
Hill Valley	GA	0.5905	1.72E−02
	BCS	0.6087	6.93E−03
	BALO	0.6426	1.32E−02
	VBPSO	0.6471	7.20E−03
	CVUBPSO	**0.6777**	1.06E−02
Urban-Land-Cover	GA	0.8748	9.63E−03
	BCS	0.8941	4.74E−03
	BALO	0.9089	8.80E−03
	VBPSO	0.9141	7.55E−03
	CVUBPSO	**0.9281**	1.15E−02

Bold indicates the best result of all the algorithms

Table 5 Number of selected features obtained by different algorithms

Dataset	Algorithm	Mean	Std
Breastcancer	GA	5.3	1.10E+00
	BCS	**5.0**	0.00E+00
	BALO	**5.0**	0.00E+00
	VBPSO	**5.0**	0.00E+00
	CVUBPSO	**5.0**	0.00E+00
HeartEW	GA	7.3	1.68E+00
	BCS	6.3	1.27E+00
	BALO	6.4	1.20E+00
	VBPSO	6.0	1.48E+00
	CVUBPSO	**5.7**	1.00E+00
IonosphereEW	GA	12.9	3.99E+00
	BCS	11.1	2.43E+00
	BALO	8.2	1.94E+00
	VBPSO	8.6	2.11E+00
	CVUBPSO	**6.6**	1.20E+00
SnoarEW	GA	28.2	3.34E+00
	BCS	28.6	3.20E+00
	BALO	25.9	5.17E+00
	VBPSO	23.8	2.68E+00
	CVUBPSO	**22.0**	2.54E+00
SpectEW	GA	8.6	2.65E+00
	BCS	10.9	1.04E+00
	BALO	9.0	1.00E+00
	VBPSO	**7.7**	1.27E+00
	CVUBPSO	9.0	7.75E−01
WaveformEW	GA	**21.4**	2.54E+00
	BCS	22.1	2.59E+00
	BALO	21.8	2.93E+00
	VBPSO	21.9	1.76E+00
	CVUBPSO	22.2	2.89E+00
PenglungEW	GA	153.6	5.54E+00
	BCS	139.4	2.50E+00
	BALO	117.5	5.70E+00
	VBPSO	101.7	5.85E+00
	CVUBPSO	**58.6**	3.49E+00
Libras movement	GA	45.3	4.45E+00
	BCS	41.1	3.94E+00
	BALO	32.8	3.74E+00
	VBPSO	32.0	2.57E+00
	CVUBPSO	**24.9**	2.69E+00
Hill Valley	GA	51.5	4.34E+00
	BCS	45.4	2.87E+00
	BALO	45.5	4.41E+00
	VBPSO	44.1	4.89E+00
	CVUBPSO	**41.5**	1.63E+00
Urban-Land-Cover	GA	74.0	7.67E+00
	BCS	67.8	5.49E+00
	BALO	63.9	3.99E+00
	VBPSO	64.5	6.33E+00
	CVUBPSO	**60.5**	5.47E+00

Bold indicates the best result of all the algorithms

Table 6 Some well known transfer functions and the proposed transfer function

Transfer function	Formula	References
TF1	$TF1(x) = erf(\frac{\sqrt{\pi}}{2}x)$	[18]
TF2	$TF2(x) = \lvert tanh(x) \rvert$	[27]
TF3	$TF3(x) = \left\lvert \frac{x}{\sqrt{1+x^2}} \right\rvert$	[22]
TF4	$TF4(x) = \lvert x^{1.5} \rvert$	[23]
TF5	$TF5(x) = \frac{1}{1+e^{-x}}$	[32]
CVUBPSO	$CVUBPSO(x) = \frac{2}{\pi}arctan(x^2)$	

only improves the quality of classification accuracy but also decreases the number of selected features significantly.

5 Conclusion

In this paper, we propose an efficient algorithm called CVUBPSO to solve the feature selection problem in different dimensions. The novel transfer function improves the exploitation ability of algorithm. The new learning strategy enhances information transmission ability of particles swarm. The local search strategy based on adaptive mutation avoid the trap of local optimization. Experiments are conducted on several classical datasets for the evaluations of the proposed algorithm, and the results show that the overall performance of CVUBPSO outperforms GA, BCS, VBPSO and BALO for solving the feature selection problem. In our future work, more test datasets will be considered to further evaluate the performance of proposed algorithm.

Table 8 The effect of the new learning strategy on fitness value

Dataset	CVUBPSO	CVUBPSO-LS
Breastcancer	**0.0269**	**0.0269**
HeartEW	0.0434	**0.0413**
IonosphereEW	0.0419	**0.0344**
SonarEW	0.0368	**0.0351**
SpectEW	0.1326	**0.1158**
WaveformEW	0.1528	**0.1499**
penglungEW	0.0023	**0.0020**
Libras Movement	0.0923	**0.0866**
Hill Valley	0.3344	**0.3274**
Urban-Land-Cover	0.0782	**0.0766**

Table 7 Comparison of using different transfer functions on BPSO

Dataset	CVUBPSO	TF1	TF2
Breastcancer	**0.0269**	0.0277	**0.0269**
HeartEW	0.0434	0.0470	**0.0431**
IonosphereEW	**0.0419**	0.0506	0.0546
SonarEW	**0.0368**	0.0473	0.0400
SpectEW	0.1326	0.1252	**0.1195**
WaveformEW	**0.1528**	0.1547	0.1583
penglungEW	**0.0023**	0.0031	0.0032
Libras Movement	**0.0923**	0.1106	0.1041
Hill Valley	**0.3344**	0.3499	0.3577
Urban-Land-Cover	**0.0782**	0.0902	0.0897
Dataset	TF3	TF4	TF5
Breastcancer	0.0274	0.0274	0.0303
HeartEW	0.0453	0.0473	0.0593
IonosphereEW	0.0506	0.0458	0.0891
SonarEW	0.0497	0.0423	0.1087
SpectEW	0.1199	0.1472	0.1695
WaveformEW	0.1562	0.1567	0.1965
penglungEW	0.0032	0.0026	0.0047
Libras Movement	0.1012	0.1049	0.1422
Hill Valley	0.3535	0.3457	0.4097
Urban-Land-Cover	0.0887	0.0834	0.1224

Bold indicates the best result of all the algorithms

Table 9 The effect of local search strategy on fitness value

Dataset	CVUBPSO-LS	CVUBPSO-LSAM
Breastcancer	0.0269	0.0269
HeartEW	0.0413	0.0410
IonosphereEW	0.0344	0.0316
SonarEW	0.0351	0.0302
SpectEW	0.1158	0.1105
WaveformEW	0.1499	0.1462
penglungEW	0.0020	0.0018
Libras Movement	0.0866	0.0811
Hill Valley	0.3274	0.3232
Urban-Land-Cover	0.0766	0.0752

Funding This work is supported in part by the National Natural Science Foundation of China (No. 62172095) and in part by Fujian University of Technology Development Fund (No. GY-Z20046)

Data availability All data used to support the findings of the study is included within this paper.

Declarations

Conflict of interest The authors declare no conflict of interest in this paper.

References

1. Al-Tashi Q et al (2019) Binary optimization using hybrid grey wolf optimization for feature selection. IEEE Access 7:39496–39508
2. Wu X et al (2020) Supervised feature selection with orthogonal regression and feature weighting. IEEE Trans Neural Netw Learn Syst 99:1–8
3. Haidi Rao AB et al (2019) Feature selection based on artificial bee colony and gradient boosting decision tree. Appl Soft Comput 74:634–642
4. Li J, Tang J, Liu H (2017) Reconstruction-based unsupervised feature selection: an embedded approach. In: Twenty-sixth international joint conference on artificial intelligence
5. Kennedy J, Eberhart R (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, 1995: MHS'95, pp 39–43. https://doi.org/10.1109/MHS.1995.494215
6. Zhao W, Zhang Z, Wang L (2020) Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications. Eng Appl Artif Intell 87:103300
7. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
8. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
9. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
10. Jia H, Peng X, Lang C (2021) Remora optimization algorithm. Expert Syst Appl 185:115665
11. Piramuthu S (2004) Evaluating feature selection methods for learning in data mining applications. Eur J Oper Res 156:483–494
12. Gunal S, Edizkan R (2008) Subspace based feature selection for pattern recognition. Inf Sci 178:3716–3726
13. Li YZ, Jiang L, Wu Q, Wang P, Gooi HB, Li K et al (2017) Wind thermal power system dispatch using MLSAD model and GSOICLW algorithm. Knowl-Based Syst 116:94–101
14. Shen Y, Wang G, Liu Q (2011) Correlative particle swarm optimization for multi-objective problems. In: Advances in swarm intelligence: second international conference, ICSI 2011, Chongqing, China, 12–15 June 2011, proceedings, Part II. Springer
15. Eberhart R, Kennedy J (2002) A new optimizer using particle swarm theory. In: Mhs95 sixth international symposium on micro machine & human science. IEEE
16. Li AD, Xue B, Zhang M (2021) Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies. Appl Soft Comput 106:107302
17. Song XF, Zhang Y, Gong DW et al (2021) Feature selection using bare-bones particle swarm optimization with mutual information. Pattern Recognit 112:107804
18. Beheshti Z, Shamsuddin SM, Hasan S (2015) Memetic binary particle swarm optimization for discrete optimization problems. Inf Sci (NY) 299:58–84
19. Shen Q et al (2004) Modified particle swarm optimization algorithm for variable selection in MLR and PLS modeling: QSAR studies of antagonism of angiotensin II antagonists. Eur J Pharm Sci 22(2–3):145–152
20. Wang L, Wang SY, Ye X (2012) An effective hybrid EDA-based algorithm for solving multidimensional knapsack problem. Expert Syst Appl 39(5):5593–5599
21. Chuang L-Y, Chang H-W, Tu C-J, Yang C-H (2008) Improved binary PSO for feature selection using gene expression data. Comput Biol Chem 32:29–38
22. Mirjalili S, Lewis A (2013) S-shaped versus V-shaped transfer functions for binary particle swarm optimization. Swarm Evol Comput 9:1–14. https://doi.org/10.1016/j.swevo.2012.09.002
23. Mirjalili S, Zhang H, Mirjalili S, Chalup S, Noman N (2020) A novel U-shaped transfer function for binary particle swarm optimisation. In: Nagar A, Deep K, Bansal J, Das K (eds) Soft Computing for Problem Solving 2019. Advances in Intelligent Systems and Computing, vol 1138. Springer, Singapore. https://doi.org/10.1007/978-981-15-3290-0_19
24. Guo S-S, Wang J-S, Guo M-W (2020) Z-shaped transfer functions for binary particle swarm optimization algorithm. Comput Intell Neurosci 2020:6502807
25. Beheshti Z (2020) A novel x-shaped binary particle swarm optimization. Soft Comput 25:3013–3042
26. Islam MJ, Li X, Mei Y (2017) A time-varying transfer function for balancing the exploration and exploitation ability of a binary PSO. Appl Soft Comput 59:182–196
27. Nezamabadi-Pour H, Rostami-Sharbabaki M, Maghfoori-Farsangi M (2008) Binary particle swarm optimization: challenges and new solutions. CSI J Comput Sci Eng 6:21–32
28. Sedighizadeh D, Masehian E, Sedighizadeh M et al (2020) GEPSO: a new generalized particle swarm optimization algorithm. Math Comput Simul 179:194–212
29. Blake CL, Merz CJ (1998) UCI repository of machine learning databases. University of California, Irvine, Irvine, CA, USA, Technical report
30. Aličković E, Subasi A (2017) Breast cancer diagnosis using GA feature selection and rotation forest. Neural Comput Appl 28(4):753–763

31. Rodrigues D, Pereira LA, Almeida T, Papa JP, Souza A, Ramos CC et al (2013) BCS: a binary cuckoo search algorithm for feature selection. In: IEEE international symposium on circuits and systems. IEEE, pp 465–468
32. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium, MHS'95. IEEE pp 39–43