Binary Sparrow Search Algorithm for Feature Selection

Xu Yuan[1], Jeng-Shyang Pan[1,3], Ai-Qing Tian[1], Shu-Chuan Chu[1,2]*

[1] *College of Computer Science and Engineering, Shandong University of Science and Technology, China*
[2] *College of Science and Engineering, Flinders University, Australia*
[3] *Department of Information Management, Chaoyang University of Technology, Taiwan*
yuanxu7772021@163.com, jspan@ieee.org, stones12138@163.com, scchu0803@gmail.com

Abstract

The sparrow search algorithm (SSA) is a novel intelligent optimization algorithm that simulates the foraging and anti-predation behavior of sparrows. The sparrow search algorithm (SSA) can optimize continuous problems, but in reality many problems are binary problems. In this paper, the binary sparrow search algorithm (BSSA) is proposed to solve binary optimization problems, such as feature selection. The transfer function is crucial to BSSA and it directly affects the performance of BSSA. This paper proposes three new transfer functions to improve the performance of BSSA. Mathematical analysis revealed that the original SSA scroungers position update equation is no longer adapted to BSSA. This paper improves the position update equation. We compared BSSA with BPSO, BGWO, and BBA algorithms, and tested on 23 benchmark functions. In addition, statistical analysis of the experimental results, Friedman test and Wilcoxon rank-sum test were performed to verify the effectiveness of BSSA. Finally, the algorithm was used to successfully implement feature selection and obtain satisfactory results in the UCI data set.

Keywords: Sparrow search algorithm, Transfer function, Benchmark function, Feature selection

1 Introduction

The meta-heuristic algorithms use empirical methods to solve some optimization problems by modeling and exploiting phenomena in nature [1-4]. The swarm intelligence optimization algorithm is part of the meta-heuristic algorithm [5-8]. The population is made up of many simple individuals, and the behavior of individuals in the group is microscopic or straightforward. Individuals in the population communicate with each other, cooperate, and there are both individual behaviors and group behaviors [9-14]. In addition, the behavior of the population will also be affected by external environments. The swarm intelligence optimization algorithms include: Particle Swarm Optimization Algorithm (PSO) to simulate bird foraging to find the optimal solution [15-18], Grey Wolf Optimization Algorithm (GWO) to simulate grey wolf foraging behavior [19-20], Sparrow Search Algorithm (SSA) to simulate sparrow foraging and anti-predation behavior [21], Whale Optimization Algorithm (WOA) to simulate whale foraging behavior [22-24], Bat Algorithm (BA) to simulate bat echolocation [25-26] and Quasi Affine Transformation Evolution Algorithm (QUATRE) [27-29].

The amount of information generated in all areas of society today has skyrocketed [30-31]. Thousands of pieces of information lead to the creation of large, high-dimensional data sets. These data sets contain a large number of features. However, not all features have classification value. In addition to valuable related features, these high-dimensional data sets also have many irrelevant, redundant, and noisy features. These irrelevant, redundant, and noisy features cause interference to other features suitable for classification, and reduce the classification ability of the data set. It is therefore essential to extract relevant features from the original features, and to remove the interference of irrelevant, redundant, and noisy features [32-34].

With the rise and development of machine learning, neural networks and data mining, feature selection has become particularly important [35-40]. By training the model with this fraction of features it is possible to obtain most of the predictive performance. Feature selection is the selection of the smallest subset of representative features from all the features that can approximately represent all the features. Feature selection is a challenging task that becomes increasingly difficult as the number of features increases. Feature selection is a typical NP-hard problem [41-43]. It is impractical to find all possible solutions to the problem by enumerating methods. The swarm intelligence optimization algorithm used to solve the combinatorial optimization problem is proved to be effective [44-47]. In recent years, studies have shown that swarm intelligence optimization algorithms are practical for feature selection. There are already binary swarm intelligence optimization algorithms applied to feature selection problems, such as the Binary Salp Swarm Algorithm (bSSA) [48], multi-objective binary genetic algorithm integrating an adaptive operator selection mechanism (MOBGA-AOS) [49], improved sticky binary PSO (ISBPSO) [50], Binary Symbiotic Organiss Search (BSOS) [51].

Feature selection is a binary problem. To solve this binary problem, the original algorithm needs to be changed to a binary version [52-54]. There are already binary versions of other algorithms used for discrete optimization problems and have achieved good results [55-56]. SSA is a new intelligent

*Corresponding Author: Shu-Chuan Chu; E-mail: scchu0803@gmail.com
DOI: 10.53106/160792642023032402001

optimization algorithm that simulates the foraging and anti-predation behavior of sparrows. This algorithm is effective in solving continuous optimization problems. This paper proposes a binary SSA algorithm for feature selection. The main contributions proposed in this paper are as follows:

(1) Three new transfer functions are proposed.

(2) The scavengers position update equation has been improved.

(3) The validity of BSSA is proved by the benchmark function.

(4) The BSSA algorithm successfully implemented feature selection in the UCI data set.

Other arrangements for the rest of the paper are as follows. Section 2 introduces the principles and implementation of the original SSA. Section 3 introduces the principle and implementation of BSSA and the three new transfer functions proposed. Section 4 verifies the effectiveness of the proposed BSSA algorithm through benchmark functions. Section 5 introduces the use of BSSA to implement feature selection on the UCI data set. Section 6 summarises the work in this paper and provides an outlook for future research.

2 Related Work

The sparrow search algorithm is a new swarm intelligence optimization algorithm introduced by J. K. Xue and B. Shen [21]. This section will introduce the basic principles and implementation of the original SSA.

2.1 Principle of the Sparrow Search Algorithm

Through years of research and observation of sparrow foraging and anti-predatory behavior researchers have discovered the mechanisms of intelligence in sparrow populations [57]. Within the sparrow population there are two identities, producers and scroungers. Producers are responsible for finding food-rich areas for sparrow populations. The identity of producers and scroungers is dynamic, but the proportion of people in the producer and the population in the population is fixed. Every sparrow has the potential to become a producer. Scavengers in sparrow populations can always spot the position of producers and move after them. The movement of sparrow populations is also influenced by predators. The sparrow will flee its current location when it realizes the danger.

2.2 Implementation of the Sparrow Search Algorithm

To make the model clearer and simpler, in the implementation of the algorithm we only consider the position information of each sparrow. Before the algorithm starts, each sparrow is randomly assigned to a location separately. The sparrow position information is shown in Eq. (1).

$$X=\begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & \cdots & X_{1,D} \\ X_{2,1} & X_{2,2} & \cdots & \cdots & X_{2,D} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{n,1} & X_{n,2} & \cdots & \cdots\vdots & X_{n,D} \end{bmatrix}. \tag{1}$$

In Eq. (1), where n is the size of the sparrow population and D is dimension. $X_{i,j}$ represents the information about the position of the i^{th} sparrow in the j^{th} dimension. The fitness of the sparrow is calculated from Eq. (2).

$$F_X=\begin{bmatrix} f([X_{1,1} & X_{1,2} & \cdots & \cdots & X_{1,D}]) \\ f([X_{2,1} & X_{2,2} & \cdots & \cdots & X_{2,D}]) \\ & & \vdots & & \\ f([X_{n,1} & X_{n,2} & \cdots & \cdots & X_{n,D}]) \end{bmatrix}. \tag{2}$$

Producers make up about 20% of the sparrow population. The update of the producers position can be represented by Eq. (3).

$$X_{i,j}^{t+1}=\begin{cases} X_{i,j}^{t}\cdot\exp(\dfrac{-i}{\alpha\cdot D}) & if\ Ra>S \\ X_{i,j}^{t}+W\cdot L1 & if\ Ra\le S \end{cases}. \tag{3}$$

In Eq. (3), t is the number of generations of i^{th} and $x_{i,j}^{t}$ is the position information of the i^{th} sparrow in the j^{th} dimension at the t iteration, j=1, 2, ... , d. i=1, 2, ... , n. D is the maximum number of iterations, which is a constant. La is a 1xd matrix where all elements are 1, W is a normally distributed random number with expectation 0 and variance 1. Ra is the alert value and S is the safety value, Ra is usually taken as 0.8 and $S \in [0.5, 1.0]$.

As soon as scroungers discover that a producer has found better position, they immediately move closer to the producer. The position update of the scavengers can be represented by Eq. (4).

$$X_{i,j}^{t+1}=\begin{cases} W\cdot\exp(\dfrac{X_{worst}^{t}-X_{i,j}^{t}}{2}) & if\ i>n/2 \\ X_{pbest}^{t+1}+\left|X_{i,j}^{t}-X_{pbest}^{t+1}\right|\cdot B^{+}\cdot La & else \end{cases}. \tag{4}$$

In Eq. (4), where X_{pbest} represents the best position currently occupied by the producers, X_{worst} is the current global worst position. La is a matrix whose values are all 1. W has the same meaning as W in Eq. (3). $B^{+} = B^{T}(BB^{T})^{-1}$, where B is a matrix of d and the elements of the matrix are all 1 or 0. When $i > n/2$, it means that these sparrows are in a poor position in the population and need to move away from the worst position. Conversely, it means that these sparrows are closer to the producer and need to follow the producer for a better source of food.

$$X_{i,j}^{t+1}=\begin{cases} X_{best}^{t}+\beta\cdot\left|X_{i,j}^{t}-X_{best}^{t}\right| & if\ f_i>f_g \\ X_{i,j}^{t}+K\cdot(\dfrac{\left|X_{i,j}^{t}-X_{worst}^{t}\right|}{(f_i-f_w)+\varepsilon}) & if\ f_i=f_g \end{cases}. \tag{5}$$

It is assumed that sparrows aware of the danger make up 20% of the total number of sparrows. The position of the aware danger sparrow is calculated from Eq. (5). where X_{best} is the current global optimum position and β is a random number with the same distribution as W in Eq. (3), which represents the step size of the move. The function of ε is to prevent the denominator from being 0, and it is a very small number. K is a random number between 0 and 1. f_i is the fitness of the i^{th} individual, f_w is the current global worst fitness, and f_g is the current global best fitness.

When $f_i > f_g$, it indicates that the sparrow aware of the danger is not the sparrow in the center of the population and that it needs to move towards its current optimal position. When $f_i = f_{gbest}$, it indicates that the sparrow in the centre of the population needs to move elsewhere. Based on the description of the above model, the pseudo-code to implement SSA is shown in Algorithm 1.

Algorithm 1. SSA pseudo code

```
Set the maximum number of iterations D to 100
Set the proportion of producers to 20%
Set the population size N to 30
Set the proportion of sparrows aware of danger to 20%
Set other related parameters
Randomly initialize the position of each sparrow
for t = 1 to D do
    Use Eq. (3) to update the producers position
    Use Eq. (4) to update the scavengers position
    Use Eq. (5) to update the position of the aware of danger sparrows
    Get the current new position
    If the new position is better than before, update it
end for
```

3 Binary Sparrow Search Algorithm

The previous section introduced the principle of the original SSA, this section will introduce the proposed BSSA. SSA has strong exploration and exploitation capabilities. Based on the above advantages of SSA, we propose a binary version of SSA.

In the original SSA, as the sparrows search space was continuous, the sparrow could theoretically reach any position in the search space. In the binary algorithm, the search space is restricted to 0 or 1. Therefore the search space of the sparrow in the binary version of the sparrow search algorithm must also be limited to 0 or 1. As there is no velocity variable in SSA, it is not possible to use velocity to implement binary in the search space as BPSO does. We have adopted a strategy to implement a binary version of SSA. Suppose each sparrow is set with a binary state tag in addition to its original position information on the continuous space, and the value of the tag is 0 or 1. Limit the search space on its continuous space to a range (in this paper, limit the position of the sparrow to [-6, 6]). Get its binary status tag by putting its position information into the transfer function. The fitness is calculated through its status tag information.

3.1 New Transfer Function

The implementation of the binary algorithm requires a transfer function. It converts a continuous value into a binary value of 0 or 1. The transfer function controls the rate of switching between 0 and 1. The efficiency of the binary algorithm is closely related to the choice of transfer function. Therefore, the selection of a suitable transfer function is crucial for the performance of the BSSA algorithm.

This paper proposes three new transfer functions. Table 1 and Figure 1 show their detailed information. Figure 2 and Figure 3 show their comparison with the original transfer function. It is known experimentally that the absolute value of x is larger at the beginning of the iteration. From Figure 2, we know that the absolute value of x of the original transfer function VO1 does not change much within the range of [2, 6], which means that the probability of position update does not change much. It is either always 0 or always 1, which is not conducive to Exploration and development within this range. In the later stages of search, the absolute value of x is small, and the original transfer function VO1 has a larger slope in the range of [0, 2], which may miss the optimal solution late in the search. The improved transfer function V1 increases the probability of updating the absolute value of x in the range [2, 6], improving the algorithm ability to claim and develop in the early stages. The slope of the [0, 2] range is reduced, reducing the probability of missing the optimal solution in the later stages of the algorithm. When x reaches a boundary value, the maximum value of the transfer function should be close to 1, so that the probability of it becoming 0 is close to 100%. However, from Figure 3 we know that when x is the boundary value, the original transfer function has a value of 0.83 and even if the function reaches its maximum value, there is a certain probability that it will not be 1. Therefore, it is essential to improve the original transfer function so that the probability of becoming 1 when it reaches the maximum value is infinitely close to 100%.

Algorithm 2. BSSA pseudo code

1: Set the maximum number of iterations D to 100
2: Set the proportion of producers to 20%
3: Set the population size N to 30
4: Set the proportion of sparrows aware of danger to 20%
5: Set other related parameters
6: Randomly initialize the position and binary tag of each sparrow
7: **for** t = 1 to D **do**
8: Use Eq. (3) to update the producers position and use transfer function update the binary tag
9: Use Eq. (6) to update the scavengers position and use transfer function update the binary tag
10: Use Eq. (5) to update the position of the aware of danger sparrows and use transfer function update the binary tag
11: Get the current new position
12: If the new position is better than before, upload it
13: **end for**

Table 1. Details of the new transfer function

Function	Function details
V1	$\left\|\tanh\left(\sqrt{\frac{\pi}{7}}x\right)\right\|$
V2	$\frac{\pi \arctan(\frac{\pi}{2}x)}{2\times\tanh(\sqrt{3})}$
V3	$\frac{5\times\tanh(\sqrt{\frac{49}{50}})}{7}\left\|\arctan(x)\right\|$

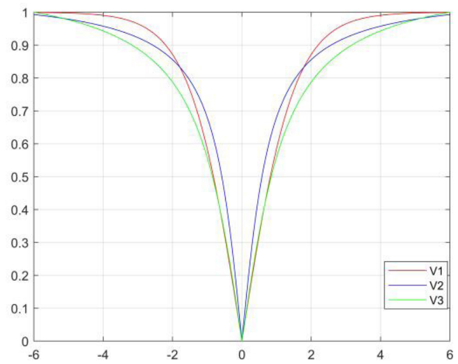

Figure 1. Three new transfer functions

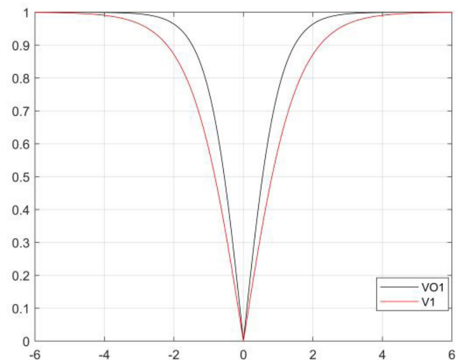

Figure 2. Compare transfer functions

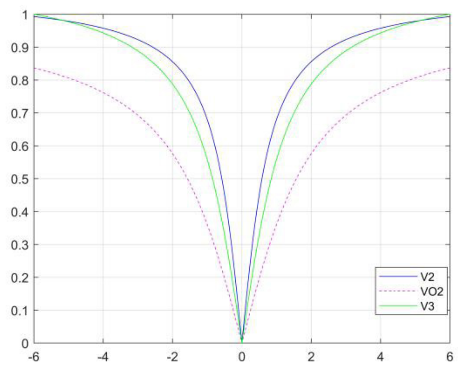

Figure 3. Compare transfer functions

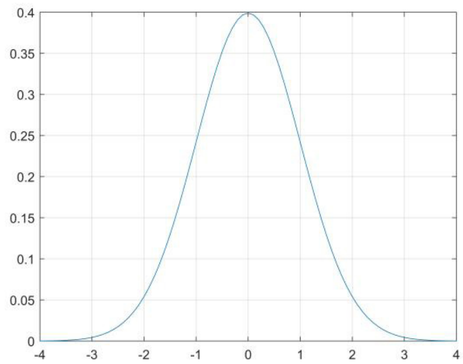

Figure 4. Standard normal distribution

Table 2. Unimodal test functions

No	Function	Space	Dim	F_{min}
F_1	$F_1(x) = \Sigma_{l=1}^{num} x_l^2$	[-100, 100]	30	0
F_2	$F_2(x) = \Sigma_{l=1}^{num} \lvert x_l \rvert + \Pi_{l=1}^{num} \lvert x_l \rvert$	[-10, 10]	30	0
F_3	$F_3(x) = \Sigma_{l=1}^{num} (\Sigma_{j=1}^{i} x_j)^2$	[-100, 100]	30	0
F_4	$F_4(x) = \max_l \{\lvert x_l \rvert, 1 \le l \le num\}$	[-100, 100]	30	0
F_5	$F_5(x) = \Sigma_{l=1}^{num-1} [100(x_{l+1} - y_l^2)^2 + (y_l - 1)^2]$	[-30, 30]	30	0
F_6	$F_6(x) = \Sigma_{l=1}^{num} ([y_l + 0.5])^2$	[-100, 100]	30	0
F_7	$F_7(x) = \Sigma_{l=1}^{num} kx_l^2 + Rand[0,1)$	[-1.28, 1.28]	30	0

3.2 Improved Updating Equation for Scavengers

In this paper the population size is 30 and the range of positional variation is [-6, 6]. Figure 4 shows that the values of the standard normal distribution are overwhelmingly distributed between [-3, 3]. We perform a mathematical analysis of the update Eq. (4) for the original position of the scavengers. When $i > n/2$, the absolute value of its position varies between approximately 0 and 3. The probability is that Sparrow will not be able to search for ranges between 3 and 6, which can significantly reduce the efficiency of the search. Therefore Eq. (4) can no longer be adapted to the sparrows search for advantage in BSSA, and an improvement to the equation is required. In order to adapt the Scavengers position update equation to the sparrows search in BSSA, we adapted the original equation for the case $i > n/2$. The adjustment strategy is to stretch it to [-6, 6] by multiplying by a factor p, which is a constant value of 2. The improved equation is Eq. (6).

$$X_{i,j}^{t+1} = \begin{cases} p \cdot W \cdot \exp(\dfrac{X_{worst}^{t} - X_{i,j}^{t}}{2}) & if\ i > n/2 \\ X_{pbest}^{t+1} + \lvert X_{i,j}^{t} - X_{pbest}^{t+1} \rvert \cdot B^{+} \cdot La & else \end{cases} . \quad \textbf{(6)}$$

3.3 Implementation of Binary Sparrow Search Algorithm

The sparrow population needs to be initialized before executing the algorithm. In BSSA not only the position of the sparrow is initialized, but also its binary tag. To be fair, the probability of the initialized label value being either 0 or 1 is set to 50% for both, as shown in Eq. (7).

$$Y_{i,j} = \begin{cases} 0 & rand() > 0.5 \\ 1 & otherwise \end{cases} . \quad \textbf{(7)}$$

In Eq. (7), $Y_{i,j}$ represents the binary tag information of the i^{th} sparrow on the j^{th} dimension, and rand() is a random number between 0 and 1.

$$Y_{i,j}^{t} = \begin{cases} 0 & F(X_{i,j}^{t}) > rand() \\ 1 & otherwise \end{cases} . \quad \textbf{(8)}$$

$$F_X = \begin{bmatrix} f([Y_{1,1} & Y_{1,2} & \cdots & \cdots & Y_{1,D}]) \\ f([Y_{2,1} & Y_{2,2} & \cdots & \cdots & Y_{2,D}]) \\ & & \vdots & & \\ f([Y_{n,1} & Y_{n,2} & \cdots & \cdots & Y_{n,D}]) \end{bmatrix} . \quad \textbf{(9)}$$

In Eq. (9), F(x) is the transfer function, and $Y_{i,j}^{t}$ is the binary state tag on the t^{th} generation, i^{th} sparrow, j^{th} dimension. We use Eq. (3), Eq. (5) and Eq. (6) to update the position of the sparrow. After updating the position each time, use Eq. (8) to get its binary status tag information. Eq. (9) is used to calculate the fitness of individuals in BSSA. The pseudo-code for BSSA is shown in Algorithm 2.

4 Simulation Experiments

In this section, the algorithm is simulated for experimentation. This paper tests the exploration capabilities of the proposed BSSA using 23 benchmark functions [18]. Table 2, Table 3, and Table 4 show the detailed information of these 23 test functions. Where F_{min}, space, and Dim represent the theoretical minimum value, search space, and dimension of the test function respectively.

The seven test functions in Table 2 are unimodal test functions, the six test functions in Table 3 are multimodal test functions, and the ten test functions in Table4 are mixed-dimensional test functions. The unimodal test function only needs to consider a global optimal solution, which can well verify the convergence performance and development ability of BSSA. Compared with the unimodal test function, the multimodal test function has multiple local optimal solutions. It can verify the global search capability of BSSA. The fixed-dimension test function has multiple local optimal solutions and multiple global optimal solutions. Use the Fixed-dimension test function to verify the performance of the BSSA more comprehensively. In order to verify the performance of BSSA, compare BSSA with other three algorithms. The details of the algorithms are shown in Table 5. The parameters of BSSA are set as follows: producers are 20% of the population size, avoiding the population clustering at a certain point. The population aware of the danger is 10%, improving the ability of the sparrow to jump out of the local optimum solution. Ra = 0.8, most producers can perform a large scale search, improving the exploration ability of the algorithm. If the population size is too large, some individuals in the population will lose their value later in the iteration, if the population size is too small, the population will lack diversity, so the population size is set to 30. The algorithm runs independently 50 times, each iteration 100 times.

4.1 Experimental Results

The experimental data is rounded to three decimal places. Table 6 shows the comparison of the mean, standard deviation (std) and minimum value (min) of the three BSSA and BPSO, BGWO, and BBA.

To demonstrate the significance of the experimental results in statistics, the Wilcoxon rank-sum test and the Friedman test were performed on the results. Table 7 shows the results of Friedman test. Table 8 shows the results of the Wilcoxon rank-sum test. The experimental data in Table 7 and Table 8 were recorded by scientific counting method.

Table 3. Multimodal test functions

No	Function	Space	Dim	F_{min}
F_8	$F_8(x)=\Sigma_{l=1}^{num}-x_l\sin(\sqrt{\lvert x_l\rvert})$	[-500, 500]	30	-12569
F_9	$F_9(x)=\Sigma_{l=1}^{num}\left[x_l^2-10\cos(2\pi x_l)+10\right]$	[-5.12, 5.12]	30	0
F_{10}	$F_{10}(x)=-20\exp(-0.2\sqrt{\frac{1}{num}\Sigma_{l=1}^{num}x_l^2})-$ $\exp(\frac{1}{num}\Sigma_{l=1}^{num}\cos(2\pi x_l)+20+2.718)$	[-32, 32]	30	0
F_{11}	$F_{11}(x)=\frac{1}{4000}\Sigma_{l=1}^{num}x_l^2-$ $\Pi_{l=1}^{num}\cos(\frac{x_l}{\sqrt{l}})+1$	[-600, 600]	30	0
F_{12}	$F_{12}(x)=\frac{\pi}{num}\{10\sin(\pi y_1)+\Sigma_{l=1}^{num-1}(y_l-1)^2$ $\left[1+10\sin^2(\pi y_{l+1})\right]+\left(y_{num}-1\right)^2\}+$ $\Sigma_{l=1}^{num}u(x_l,10,100,4)y_l=1+\frac{x_l+1}{4}$ $u(x_l,a,k,r)=$ $\begin{cases}k(x_l-a)^r & x_l>a\\ 0 & -a<x_l<a\\ k(-x_l-a)^r & x_l<-a\end{cases}$	[-50, 50]	30	0
F_{13}	$F_{13}(x)=0.1\{\sin^2(3\pi x_1)+$ $\Sigma_{l=1}^{num}(x_l-1)^2\left[1+\sin^2(3\pi x_l+1)\right]$ $+(x_{num}-1)^2\left[1+\sin^2(2\pi x_{num})\right]\}+$ $\Sigma_{l=1}^{num}u(x_l,10,100,4)$	[-50, 50]	30	0

Table 4. Fixed-dimension test functions

No	Function	Space	Dim	F_{min}
F_{14}	$F_{14}(x)=\left(\frac{1}{500}\Sigma_{l=1}^{25}\frac{1}{l+\Sigma_{j=1}^{2}(z_j-a_{jl})}\right)^{-1}$	[-65, 65]	2	1
F_{15}	$F_{15}(x)=\Sigma_{l=1}^{11}\left[a_l-\frac{x_1(c_l^2+b_l x_2)}{c_l^2+c_l x_3+x_4}\right]^2$	[-5, 5]	4	0.0003
F_{16}	$F_{16}(x)=4x_l^2-2.1x_l^4+\frac{1}{3}x_l^6+x_2x_i-4_2^2+4x_2^4$	[-5, 5]	2	-1.0316
F_{17}	$F_{17}(x)=\left(x_2-\frac{5.1}{4\pi^2}x_l^2+\frac{5}{\pi}x_l-6\right)^2+$ $10(1-\frac{1}{8\pi})\cos x_l+10$	[-5, 5]	2	0.398
F_{18}	$F_{18}(x)=\left[1+(x_1+x_2+l)^2\times(19-14x_1+3x_1^2-\right.$ $\left.14x_2+6x_2x_1+3x_2^2)\times\right]\left[30+(2x_1-3x_2)^2\times(18\right.$ $\left.-32x_1+12x_1^2+48x_2-36x_2x_1+27x_2^2)\right]$	[-2, 2]	2	3
F_{19}	$F_{19}(x)=-\Sigma_{i=1}^{4}c_l\times\exp\left(-\Sigma_{l=1}^{3}a_{lj}\left(x_j-p_{lj}\right)^2\right)$	[1, 3]	3	-3.86
F_{20}	$F_{20}(x)=-\Sigma_{i=1}^{4}c_l\times\exp\left(-\Sigma_{j=1}^{6}a_{lj}\left(x_j-p_{lj}\right)^2\right)$	[0, 1]	6	-3.32
F_{21}	$F_{21}(x)=-\Sigma_{l=1}^{5}\left[\left(x-a_l\right)\left(x-a_l\right)^M+c_l\right]^{-1}$	[0, 10]	4	-10.1532
F_{22}	$F_{22}(x)=-\Sigma_{l=1}^{7}\left[\left(x-a_l\right)\left(x-a_l\right)^M+c_l\right]^{-1}$	[0, 10]	4	-10.4028
F_{23}	$F_{23}(x)=-\Sigma_{l=1}^{10}\left[\left(x-a_l\right)\left(x-a_l\right)^M+c_l\right]^{-1}$	[0, 10]	4	-10.5363

Table 5. The details of the algorithm

Name	Transfer function
BSSA-V1	$\left\lvert\tanh(\sqrt{\frac{\pi}{7}}x)\right\rvert$
BSSA-V2	$\frac{\pi\arctan(\frac{\pi}{2}x)}{2\times\tanh(\sqrt{3})}$
BSSA-V3	$\frac{5\times\tanh(\sqrt{\frac{49}{50}})}{7}\lvert\arctan(x)\rvert$

Table 6. Performance comparison of the algorithm on 23 test functions

Function	BPSO			BGWO			BBA		
	MEAN	STD	MIN	MEAN	STD	MIN	MEAN	STD	MIN
F_1	4.980	0.769	3.000	3.060	1.300	0.000	0.000	0.000	0.000
F_2	5.120	0.799	3.000	2.980	1.152	1.000	0.000	0.000	0.000
F_3	205.520	64.162	58.000	84.680	78.341	1.000	0.000	0.000	0.000
F_4	1.000	0.000	1.000	1.000	0.000	1.000	1.000	0.000	1.000
F_5	489.180	64.117	406.000	22.380	55.335	0.000	0.000	0.000	0.000
F_6	17.780	1.565	13.500	13.740	2.576	7.500	7.500	0.000	7.500
F_7	61.497	10.269	31.543	32.100	18.805	0.000	0.001	0.001	0.000
F_8	-21.003	0.679	-22.720	-25.244	0.000	-25.244	-25.244	0.000	-25.244
F_9	5.040	0.755	3.000	3.320	1.115	0.000	0.000	0.000	0.000
F_{10}	1.562	0.099	1.409	1.226	0.283	0.717	0.000	0.000	0.000
F_{11}	0.162	0.025	0.106	0.145	0.049	0.058	0.000	0.000	0.000
F_{12}	2.462	0.133	2.121	2.190	0.227	1.669	1.669	0.000	1.669
F_{13}	0.500	0.078	0.300	0.000	0.000	0.000	0.000	0.000	0.000
F_{14}	12.671	0.000	12.671	12.671	0.000	12.671	12.671	0.000	12.671
F_{15}	0.148	0.000	0.148	0.148	0.000	0.148	0.148	0.000	0.148
F_{16}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
F_{17}	27.703	0.000	27.703	27.703	0.000	27.703	27.703	0.000	27.703
F_{18}	600.000	0.000	600.000	600.000	0.000	600.000	600.000	0.000	600.000
F_{19}	-0.335	0.000	-0.335	-0.334	0.005	-0.335	-0.335	0.000	-0.335
F_{20}	-0.166	0.000	-0.166	-0.147	0.036	-0.166	-0.166	0.000	-0.166
F_{21}	-5.055	0.000	-5.055	-5.055	0.000	-5.055	-5.055	0.000	-5.055
F_{22}	-5.088	0.000	-5.088	-5.088	0.000	-5.088	-5.088	0.000	-5.088
F_{23}	-5.128	0.000	-5.128	-5.128	0.000	-5.128	-5.128	0.000	-5.128

Table 6 (continued). Performance comparison of the algorithm on 23 test functions

Function	BSSA-V1			BSSA-V2			BSSA-V3		
	MEAN	STD	MIN	MEAN	STD	MIN	MEAN	STD	MIN
F_1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
F_2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
F_3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
F_4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
F_5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
F_6	7.500	0.000	7.500	7.500	0.000	7.500	7.500	0.000	7.500
F_7	0.002	0.002	0.000	16.385	7.455	0.692	0.013	0.038	0.000
F_8	-25.244	0.000	-25.244	-25.244	0.000	-25.244	-25.244	0.000	-25.244
F_9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
F_{10}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
F_{11}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
F_{12}	1.669	0.000	1.669	1.669	0.000	1.669	1.669	0.000	1.669
F_{13}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
F_{14}	12.671	0.000	12.671	12.671	0.000	12.671	12.671	0.000	12.671
F_{15}	0.148	0.000	0.148	0.148	0.000	0.148	0.148	0.000	0.148
F_{16}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
F_{17}	27.703	0.000	27.703	27.703	0.000	27.703	27.703	0.000	27.703
F_{18}	600.000	0.000	600.000	600.000	0.000	600.000	600.000	0.000	600.000
F_{19}	-0.335	0.000	-0.335	-0.335	0.000	-0.335	-0.335	0.000	-0.335
F_{20}	-0.166	0.000	-0.166	-0.166	0.000	-0.166	-0.166	0.000	-0.166
F_{21}	-5.055	0.000	-5.055	-5.055	0.000	-5.055	-5.055	0.000	-5.055
F_{22}	-5.088	0.000	-5.088	-5.088	0.000	-5.088	-5.088	0.000	-5.088
F_{23}	-5.128	0.000	-5.128	-5.128	0.000	-5.128	-5.128	0.000	-5.128

Table 7. Freedman test results

	SS	DF	P
F_1	888	349	9.65E-61
F_2	898.5	349	2.46E-61
F_3	900	349	3.59E-61
F_4	1050	349	8.18E-62
F_5	685	349	8.81E-50
F_6	890	349	7.71E-61
F_7	1400	349	2.92E-49
F_8	525	349	8.18E-62
F_9	889.5	349	8.20E-61
F_{10}	897.5	349	4.43E-61
F_{11}	900	349	3.61E-60
F_{12}	885	349	4.84E-60
F_{13}	525	349	8.18E-62
F_{14}	0	349	1
F_{15}	0	349	1
F_{16}	0	349	1
F_{17}	0	349	1
F_{18}	0	349	1
F_{19}	10.5	349	0.4232
F_{20}	157.5	349	3.03E-17
F_{21}	0	349	1
F_{22}	0	349	1
F_{23}	0	349	1

Table 8. Wilcoxon rank-sum test results

	BSSA-V1			BSSA-V2			BSSA-V3		
	BPSO	BGWO	BBA	BPSO	BGWO	BBA	BPSO	BGWO	BBA
F_1	1.28E-20	2.26E-20	2.26E-20	1.09E-20	9.21E-20	-	1.09E-20	9.21E-20	-
F_2	1.28E-20	2.26E-20	-	1.28E-20	2.26E-20	-	1.28E-20	2.26E-20	-
F_3	3.30E-20	3.30E-20	-	3.30E-20	3.30E-20	-	3.30E-20	3.30E-20	-
F_4	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23	2.63E-23
F_5	3.28E-20	0.0035	-	3.28E-20	0.0035	-	3.28E-20	0.0035	-
F_6	1.25E-20	9.16E-20	-	1.25E-20	9.16E-20	-	1.25E-20	9.16E-20	-
F_7	7.07E-18	2.99E-05	7.07E-18	7.07E-18	1.35E-16	0.2237	7.07E-18	1.07E-16	0.0235
F_8	1.10E-20	-	-	1.10E-20	-	-	1.10E-20	-	-
F_9	1.26E-20	7.74E-20	-	1.26E-20	7.74E-20	-	1.26E-20	7.74E-20	-
F_{10}	8.60E-21	2.52E-20	-	8.60E-21	2.52E-20	-	8.60E-21	2.52E-20	-
F_{11}	3.31E-20	3.31E-20	-	3.31E-20	3.31E-20	-	3.31E-20	3.31E-20	-
F_{12}	3.04E-20	4.48E-19	-	3.04E-20	4.48E-19	-	3.04E-20	4.48E-19	-
F_{13}	8.76E-21	-	-	8.76E-21	-	-	8.76E-21	-	-
F_{14}	-	-	-	-	-	-	-	-	-
F_{15}	-	-	-	-	-	-	-	-	-
F_{16}	-	-	-	-	-	-	-	-	-
F_{17}	-	-	-	-	-	-	-	-	-
F_{18}	-	-	-	-	-	-	-	-	-
F_{19}	-	0.3271	-	-	0.3271	-	-	0.3271	-
F_{20}	-	3.18E-05	-	-	3.18E-05	-	-	3.18E-05	-
F_{21}	-	-	-	-	-	-	-	-	-
F_{22}	-	-	-	-	-	-	-	-	-
F_{23}	-	-	-	-	-	-	-	-	-

Table 9. The time consumption of the BSSA

	BSSA-V1	BSSA-V2	BSSA-V3
F_1	0.0779	0.0795	0.0761
F_2	0.0763	0.0768	0.0776
F_3	0.1124	0.1072	0.1087
F_4	0.0921	0.0788	0.0757
F_5	0.0819	0.0782	0.0810
F_6	0.0819	0.0755	0.0764
F_7	0.0825	0.0774	0.0782
F_8	0.0889	0.0778	0.0780
F_9	0.0835	0.0796	0.0849
F_{10}	0.0834	0.0857	0.0746
F_{11}	0.0890	0.0836	0.0790
F_{12}	0.1443	0.1281	0.1228
F_{13}	0.1456	0.1257	0.1203
F_{14}	0.1301	0.1133	0.1118
F_{15}	0.0458	0.0369	0.0352
F_{16}	0.0332	0.0294	0.0292
F_{17}	0.0332	0.0294	0.0290
F_{18}	0.0305	0.0299	0.0296
F_{19}	0.0407	0.0359	0.0349
F_{20}	0.0446	0.0397	0.0386
F_{21}	0.0472	0.0399	0.0385
F_{22}	0.0471	0.0410	0.0399
F_{23}	0.0463	0.0440	0.0433

Table 10. Details of the data set

Datasets	Instances	Number of class-es	Number of features	Size of classes
Wine	178	3	13	59,71,48
Wdbc	569	2	30	357,212
WBC	683	2	9	444,239
Vowel	871	6	4	72,89,172,151,207,180
Thyroid	215	3	5	150,35,30
Sonar	208	2	60	97,111
Seeds	210	3	7	70,70,70
Jain	373	2	2	276,97
Ionosphere	351	2	34	126,225
Diabetes	768	2	8	268,500
CMC	1473	3	9	629,333,511
Bupa	345	2	6	145,200
Aggregation	788	7	2	170,34,273,102,130,45,34
Appendicitis	106	2	7	21,85
Austra	690	2	14	222,468
Australian	690	2	14	222,468
Breast	277	2	9	196,81
Ecoli_label	335	2	2	223,112
Glass_gy	214	6	9	29,76,70,17,13,9
Segmentation	210	7	18	30,30,30,30,30,30,30
Weather	22	2	4	10,22

Table 11. Comparison of fitness

Datasets	BSSA-V1	BSSA-V2	BSSA-V3	BPSO	BGWO	BBA
Wine	0.012692308	0.011015385	0.013280769	0.019319231	0.015461538	0.033969231
Wdbc	0.028889872	0.028725	0.028148205	0.032120897	0.032943846	0.037876282
WBC	0.021360106	0.023264273	0.02179016	0.025361879	0.026805437	0.027860047
Vowel	0.135172787	0.133571475	0.135838197	0.139392459	0.135156557	0.152385628
Thyroid	0.04004	0.04004	0.03912	0.05312	0.05088	0.06012
Sonar	0.116984762	0.112779048	0.108640952	0.129357143	0.1309	0.154370476
Seeds	0.041231429	0.041514286	0.040574286	0.050671429	0.053428571	0.065114286
Jain	0.01	0.01	0.01	0.01	0.01	0.016587179
Ionosphere	0.001428571	0.000742857	0.001428571	0.017523571	0.035007857	0.043043
Diabetes	0.214384277	0.215338994	0.217869182	0.223822956	0.219611006	0.239996541
CMC	0.474872494	0.470396702	0.472666115	0.493217619	0.492797467	0.502314397
Bupa	0.298597667	0.29607	0.297030333	0.314979	0.301492	0.349173667
Aggregation	0.01	0.010713571	0.01	0.01072	0.010353571	0.098807143
Appendicitis	0.89685	0.897910714	0.900453571	0.918064286	0.912925	0.934546429
Austra	0.30201352	0.310119133	0.307104847	0.333241837	0.332656378	0.349002296
Australian	0.290336735	0.287819898	0.288424745	0.310589031	0.316669898	0.328214796
Breast	0.514968713	0.509681579	0.520173041	0.548138363	0.556144269	0.587008889
Ecoli_label	0.194140217	0.194068478	0.194283696	0.194427174	0.194176087	0.19818913
Glass_gy	0.964051111	0.964051111	0.964051111	0.970388254	0.979163492	0.978065714
Segmentation	0.904697895	0.898308421	0.901075639	0.960173383	0.961984812	0.967615489
Weather	0.2995	0.2995	0.3094	0.3307	0.2916	0.4505

4.2 Analysis of Experimental Results

Among the seven unimodal test functions, the proposed BSSA performs quite well and can basically reach or approach the theoretical optimal value stably. BBA has the closest performance to BSSA, while BPSO and BGWO perform poorly. Table 6 shows that in the F_1- F_5 test function, BSSA can reach the theoretical optimal value stably. Although F_6 and F_7 cannot reach the theoretical minimum value, they are closer to the minimum value than the result of the other algorithms. The performance of BBA in the unimodal test function is basically the same as that of BSSA except for F_4.

Among the six multimodal test functions, the proposed BSSA can reach the theoretical minimum value at F_8, F_9 and F_{11}. BGWO and BPSO do not reach the theoretical optimum in all of the multimodal test functions and deviate from the optimum by a large margin. Although BSSA did not give the best theoretical results for the remaining multimodal test functions, the difference between them was small.

Table 6 shows that these algorithms performed approximately the same in the remaining test functions. In the Fixed-dimension test function, the difference of the algorithm is not obvious. This is because in the case of low dimensionality, the difference in performance between algorithms is not easy to find due to the impact of dimensionality. The above experimental results show that the BSSA outperforms the three comparison algorithms on the twenty-three test functions. This means that the improved position update equation in this paper for the binary version of the SSA algorithm are valid. As well as demonstrating that the three new transfer functions proposed have excellent transformation capabilities, which results in a stronger performance of BSSA. The performance of BSSA in the Fixed-dimension test function did not meet the expected results, but its overall results are acceptable.

Table 7 shows the results of the Friedman test. Except for F_{14}- F_{19}, F_{21}- F_{23} the p-values for the rest of the samples are less than 0.05, so the original hypothesis is overturned. The results of the rank-sum test in Table 8 also illustrate this point.

The time complexity of the original sparrow search algorithm is O($T * dim * (N * (F) + EQT)$). The binary sparrow search algorithm adds the calculation of the transfer function, and the time complexity of the binary sparrow search algorithm is O($T *$ dim $(N * (F + TF) + EQT)$), where T is the number of iteration，dim is the dimension，N represents the population size, F implies the time consumed to calculate the fitness, TF indicates the time consumed by the transfer function，EQT is the time consumed to update the equation. Table 9 indicates the average running time of the algorithm in seconds. BSSA-V1, BSSA-V2 and BSSA-V3 do not differ significantly in average running time. This is because only the transfer functions differ for the proposed algorithms.

5 Application to Feature Selection

In most machine learning tasks, the selection of features determines the upper limit of the learning algorithm. However, among the many features, which ones are valuable and which ones are redundant are unknown. These redundant features will lead to inefficient learning algorithms. Feature selection is the process of extracting valuable features from all the features to improve the efficiency of the learning algorithm. Feature selection is significant for machine learning.

5.1 Feature Selection

In reality, an object usually has many features. Irrelevant and redundant features will interfere with the classification and reduce the efficiency of the learning algorithm. The task of feature selection is to select a subset of features with as few features as possible, the effect of the model will not decrease significantly, and the category distribution of the result is as close to the real situation as possible. Feature selection mainly includes four processes: (1). Generation process. (2). Evaluation function. (3). Stop condition. (4). Verification process.

5.2 Data Set Description

21 data sets were selected for the experiment. These datasets are from UCI Machine Learning [58] and they have different attributes and examples, as detailed in Table 10.

The K-nearest neighbour (KNN) classification algorithm is one of the most widely used methods in data mining classification techniques [59]. Its guiding ideology is to infer your category by your neighbors. It works by using samples that have been correctly classified as a reference to classify samples from unknown categories. The unknown samples and the K nearest known samples were grouped into one category.

$$Dist\left(x^{(1)},x^{(2)}\right)=\sqrt{\sum_{l=1}^{n}\left(x^{(1)}(l)-x^{(2)}(l)\right)^{2}}\ . \tag{10}$$

$$Dist\left(x^{(1)},x^{(2)}\right)=\sum_{l=1}^{n}\left|x^{(1)}(l)-x^{(2)}(l)\right|\ . \tag{11}$$

According to the characteristics of distance in the KNN algorithm, the most commonly used calculation equations are Euclidean distance and Manhattan distance. Eq. (10) is the equation for the Euclidean distance and Eq. (11) is the equation for the Manhattan distance. In Eq. (10) and Eq. (11), $x^{(1)}$ is the training set, $x^{(2)}$ is the test set, n represents the number of features, and i represents the i^{th} feature.

Cross-validation is a commonly used statistical method for generalization capability of a data set, which is mainly used to prevent over-fitting caused by overly complex models. In machine learning, the number of data set samples is not always sufficient. The training data were randomly divided into K groups and any one of them was chosen as the test set and the rest as the training set. K-fold validation is not repeated for each sample selection, which ensures that each sample point is assigned to the training set or the test machine only once in each iteration.

Table 12. Comparison of accuracy rates

Datasets	BSSA-V1	BSSA-V2	BSSA-V3	BPSO	BGWO	BBA
Wine	0.992307692	0.993846154	0.992179487	0.985769231	0.971282051	0.971282051
Wdbc	0.975397436	0.975833333	0.977897436	0.973884615	0.966858974	0.966858974
WBC	0.982464539	0.981214539	0.982030142	0.979095745	0.977021277	0.977021277
Vowel	0.873562842	0.875180328	0.87289071	0.869300546	0.855502732	0.855502732
Thyroid	0.964	0.964	0.965333333	0.952	0.945333333	0.945333333
Sonar	0.858761905	0.851619048	0.863714286	0.821142857	0.818857143	0.818857143
Seeds	0.961238095	0.960952381	0.962190476	0.952857143	0.94	0.94
Jain	1	1	1	1	1	1
Ionosphere	1	1	1	0.986166667	0.960966667	0.960966667
Diabetes	0.789510832	0.788546471	0.785485674	0.778714186	0.763892383	0.763892383
CMC	0.524146751	0.52821882	0.525702016	0.50651419	0.497774459	0.497774459
Bupa	0.7031	0.707	0.706366667	0.6879	0.654033333	0.654033333
Aggregation	1	0.999279221	1	0.999272727	0.909285714	0.909285714
Appendicitis	0.099285714	0.098214286	0.095357143	0.077857143	0.061785714	0.061785714
Austra	0.69710034	0.689634354	0.692967687	0.667721088	0.651079932	0.651079932
Australian	0.709183673	0.711581633	0.711692177	0.69002551	0.673520408	0.673520408
Breast	0.483421053	0.489210526	0.476368421	0.451263158	0.412	0.412
Ecoli_label	0.808949275	0.809021739	0.808804348	0.80865942	0.805869565	0.805869565
Glass_gy	0.027333333	0.027333333	0.027333333	0.022952381	0.016095238	0.016095238
Segmentation	0.087333333	0.094	0.09152381	0.034380952	0.027714286	0.027714286
Weather	0.7	0.7	0.69	0.67	0.55	0.55

Table 13. Comparison of feature number

Datasets	BSSA-V1	BSSA-V2	BSSA-V3	BPSO	BGWO	BBA
Wine	6.6	6.4	7.2	6.8	7.2	7.2
Wdbc	13.6	14.4	18.8	18.8	15.2	15.2
WBC	3.6	4.2	3.6	4.2	4.6	4.6
Vowel	3	3	3	3	2.8	2.8
Thyroid	2.2	2.2	2.4	2.8	3	3
Sonar	18	17.6	17	29	32.6	32.6
Seeds	2	2	2.2	2.8	4	4
Jain	2	2	2	2	2	2
Ionosphere	5	2.6	5	13.4	15.4	15.4
Diabetes	4.8	4.8	4.4	3.8	5	5
CMC	3.4	3	2.8	4.2	4.6	4.6
Bupa	2.8	3.6	3.8	3.6	4	4
Aggregation	2	2	2	2	1.8	1.8
Appendicitis	3.6	3.6	3.4	3.6	4	4
Austra	3	4	4.4	6	5	5
Australian	3.4	3.2	4.2	5.2	7	7
Breast	3.2	3.6	1.6	4.4	4.4	4.4
Ecoli_label	1	1	1	1	1.2	1.2
Glass_gy	1	1	1	2.8	3.6	3.6
Segmentation	2.2	2.6	3.2	8	9.6	9.6
Weather	1	1	1	1.6	2	2

The result of feature selection is not only the accuracy but also the number of features. The aim is to get a high accuracy rate with a low number of features by feature selection. Therefore, the evaluation standard should take into account both of these indicators. The evaluation function used in this paper is Eq. (12).

$$Fit = c1 \times err + c2 \times \frac{|S|}{|AL|}. \tag{12}$$

In Eq. (12), *err* is the number of errors in the classification as a proportion of the total, *S* is the set of extracted features, *AL* denotes the sum of the features to be extracted in the data set, and *c1* and *c2* are coefficients, 0,99 and 0.01 respectively.

5.3 Analysis of Experimental Results

In the experimental simulation, BSSA was compared with BGWO, BPSO and BBA respectively. The algorithm runs independently on each data set 20 times, the number of population iterations is set to 100, and the population size is set to 30.

Table 11 shows the statistical results of fitness calculated by Eq. (12). It can be seen from Table 11 that BSSA-V1, BSSA-V2, and BSSA-V3 are basically better than the three algorithms compared. It can be seen from the statistics of fitness in Table 11 that BSSA-V1 has 18 better than comparison algorithms, BSSA-V2 has 19 better than comparison algorithms, and BSSA-V3 has 19 better than comparison algorithms.

Table 12 and Table 13 respectively show the accuracy and the number of selected features. Mark the ones that are better than all comparison algorithms in the Table in red. Table 12 and Table 13 shows that BSSA performs better than BPSO, BGWO, and BBA in most data sets. BSSA selects fewer features in the case of higher accuracy. In the Jain data set, BSSA-V1, BSSA-V2, BSSA-V3 did not perform differently from the comparison algorithm. This may be due to the small size of the categories in the Jain data set resulting in insignificant differences. The difference between BSSA-V2 and BPSO in the Aggregation data set is not obvious, and the reason may also be that the data set has fewer categories. The experimental results show that BSSA is applied to feature selection is effective.

6 Conclusions

This paper proposes a binary version of SSA to solve the binary problem. Through the image analysis of transfer functions, this paper proposes three new transfer functions. In addition, mathematical analysis knows that the original position update equation can no longer be adapted to the binary version of the algorithm, which is improved in this paper. In order to verify the performance of BSSA, three types of benchmark functions were selected for testing and comparing with BPSO, BGWO and BBA. The experimental results prove that the performance of BSSA is better than the other three comparison algorithms. We also performed Friedman test and rank-sum test to further verify the performance of BSSA. Finally, we applied the BSSA algorithm to feature selection. Experimental results show that BSSA can maintain good classification accuracy while choosing fewer features. The algorithm will be applied to more fields in future research, such as knapsack problem and traveling salesman problem.

References

[1] P. C. Song, S.-C. Chu, J.-S. Pan, H. M. Yang, Simplified Phasmatodea population evolution algorithm for optimization, *Complex & Intelligent Systems*, pp. 1-19 June, 2021.

[2] M. Mavrovouniotis, C. Li, S. Yang, A survey of swarm intelligence for dynamic optimization: Algorithms and applications, *Swarm and Evolutionary Computation*, Vol. 33, pp. 1-17, April, 2017.

[3] X. Kou, J. Feng, Matching Ontologies through Compact Monarch Butterfly Algorithm, *Journal of Network Intelligence*, Vol. 5, No. 4, pp. 191-197, November, 2020.

[4] E. Hopper, B. C. H. Turton, An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem, *European Journal of Operational Research*, Vol. 128, No. 1, pp. 34-57, January, 2001.

[5] X. S. Yang, Flower Pollination Algorithm for Global Optimization, in: J. Durand-Lose, N. Jonoska (Eds.), *Lecture Notes in Computer Science*, Vol. 7445, Springer, Berlin, Heidelberg, 2012, pp. 240-249.

[6] S. Chakraborty, A. K. Saha, R. Chakraborty, M. Saha, An enhanced whale optimization algorithm for large scale optimization problems, *Knowledge-Based Systems*, Vol. 233, pp. 107543, December, 2021.

[7] Z. Barta, A. Liker, F. Monus, The effects of predation risk on the use of social foraging tactics, *Animal Behaviour*, Vol. 67, No. 2, pp. 301-308, February, 2004.

[8] M. Dorigo, G. D. Caro, Ant colony optimization: a new meta-heuristic, *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Vol. 2, Washington, DC, USA, 1999, pp. 1470-1477.

[9] H. Robles-Berumen, A. Zafra, H. M. Fardoun, S. Ventura, LEAC: An efficient library for clustering with evolutionary algorithms, *Knowledge-Based Systems*, Vol. 179, pp. 117-119, September, 2019.

[10] J.-S. Pan, T. T. Nguyen, T. K. Dao, T. S. Pan, S.-C. Chu, Clustering Formation in Wireless Sensor Networks: A Survey, *Journal of Network Intelligence*, Vol. 2, No. 4, pp. 287-309, November, 2017.

[11] T. K. Dao, T. S. Pan, J.-S. Pan, A multi-objective optimal mobile robot path planning based on whale optimization algorithm, *13th IEEE International Conference on Signal Processing (ICSP)*, Chengdu, China, 2016, pp. 337-342.

[12] X. P. Wang, J.-S. Pan, S.-C. Chu, A Parallel Multi-Verse Optimizer for Application in Multilevel Image Segmentation, *IEEE Access*, Vol. 8, pp. 32018-32030, February, 2020.

[13] J.-S. Pan, Z. Y. Meng, S.-C. Chu, H. R. Xu, Monkey King Evolution: an enhanced ebb-tide-fish algorithm for global optimization and its application in vehicle navigation under wireless sensor network environment, *Telecommunication Systems*, Vol. 65, No. 3, pp. 351-364, July, 2017.

[14] H. C. Huang, S.-C. Chu, J.-S. Pan, C. Y. Huang, B. Y. Liao, Tabu Search Based Multi-watermarks Embedding Algorithm with Multiple Description Coding, *Information Sciences*, Vol. 181, No. 16, pp. 3379-3396, August, 2011.

[15] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proceedings of ICNN'95-International Conference on Neural Networks*, Vol. 4, Perth, WA, Australia, 1995, pp. 1942-1948.

[16] M. S. Kiran, Particle swarm optimization with a new update mechanism, *Applied Soft Computing*, Vol. 60, pp. 670-678, November, 2017.

[17] S. Mirjalili, A. Lewis, S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization, *Swarm and Evolutionary Computation*, Vol. 9, pp. 1-14, April, 2013.

[18] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization, *Swarm intelligence*, Vol. 1, No. 1, pp. 33-57, June, 2007.

[19] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Advances in Engineering Software*, Vol. 69, pp. 46-61, March, 2014.

[20] H. Faris, L. Aljarah, M. A. Al-Betar, S. Mirjalili, Grey wolf optimizer: a review of recent variants and applications, *Neural Computing and Applications*, Vol. 30, No. 2, pp. 413-435, July, 2018.

[21] J. K. Xue, B. Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, *Systems Science & Control Engineering*, Vol. 8, No. 1, pp. 22-34, 2020.

[22] J. Nasiri, F. M. Khiyabani, A. Yoshise, A whale optimization algorithm (WOA) approach for clustering, *Cogent Mathematics & Statistics*, Vol. 5, No. 1, Article No. 1483565, 2018.

[23] S. Mirjalili, A. Lewis, The Whale Optimization Algorithm, *Advances in Engineering Software*, Vol. 95, pp. 51-67, May, 2016.

[24] H. L. Chen, Y. T. Xu, M. J. Wang, X. H. Zhao, A balanced whale optimization algorithm for constrained engineering design problems, *Applied Mathematical Modelling*, Vol. 71, pp. 45-59, July, 2019.

[25] A. H. Gandomi, X. S. Yang, A. H. Alavi, S. Talatahari, Bat algorithm for constrained optimization tasks, *Neural Computing and Applications*, Vol. 22, No. 6, pp. 1239-1255, May, 2013.

[26] S. Yilmaz, E. U. Kucuksille, Y. Cengiz, Modified bat algorithm, *Elektronika ir Elektrotechnika*, Vol. 20, No. 2, pp.71-78, 2014.

[27] N. X. Liu, J.-S Pan, S.-C. Chu, A Competitive Learning QUasi Affine TRansformation Evolutionary for Global Optimization and Its Application in CVRP, *Journal of Internet Technology*, Vol. 21, No. 7, pp. 1863-1883, December, 2020.

[28] Z. Meng, J.-S. Pan, QUasi-Affine TRansformation Evolution with External ARchive (QUATRE-EAR): an enhanced structure for differential evolution,

Knowledge-Based Systems, Vol. 155, pp. 35-53, September, 2018.

[29] S.-C. Chu, Z. J. Zhuang, J. B. Li, J.-S. Pan, A Novel Binary QUasi-Affine TRansformation Evolutionary (QUATRE) Algorithm, *Applied Sciences*, Vol. 11, No. 5, Article No. 2251, March, 2021.

[30] B. Xue, M. J. Zhang, W. N. Browne, Particle swarm optimization for feature selection in classification: A multi-objective approach, *IEEE transactions on cybernetics*, Vol. 43, No. 6, pp. 1656-1671, December, 2013.

[31] F. L. Chen, F. C. Li, Combination of feature selection approaches with SVM in credit scoring, *Expert systems with applications*, Vol. 37, No. 7, pp. 4902-4909, July, 2010.

[32] A. L. Marques, V. Garcia, J. S. Sanchez, Exploring the behaviour of base classifiers in credit scoring ensembles, *Expert Systems with Applications*, Vol. 39, No. 11, pp. 10244-10250, September, 2012.

[33] S. Yu, W. M. Tan, C. M. Zhang. Y. Fang. C. Tang. D. Hu, Research on hybrid feature selection method of power transformer based on fuzzy information entropy, *Advanced Engineering Informatics*, Vol. 50, Article No. 101433, October, 2021.

[34] J. Cai, J. W. Luo, S. L. Wang, S. Yang, Feature selection in machine learning: A new perspective, *Neurocomputing*, Vol. 300, pp. 70-79, July, 2018.

[35] Z. Chen, M. Pan, Z. X. Zhao, S. N. Li, R. Miao, Y. F. Zhang, X. Y. Feng, X. Feng, Y. X. Zhang, M. Y. Duan, L. Huang, F. F. Zhou, Feature selection may improve deep neural networks for the bioinformatics problems, *Bioinformatics*, Vol. 36, No. 5, pp. 1542-1552, March, 2020.

[36] S. C. Yusta, Different metaheuristic strategies to solve the feature selection problem, *Pattern Recognition Letters*, Vol. 30, No. 5, pp. 525-534, April, 2009.

[37] L. Y. Yang. Z. S. Xu, Feature extraction by PCA and diagnosis of breast tumors using SVM with DE-based parameter tuning, *International Journal of Machine Learning and Cybernetics*, Vol. 10, No. 3, pp. 591-601, March, 2019.

[38] B. Agarwal, N. Mittal, Prominent feature extraction for review analysis: an empirical study, *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 28, No. 3, pp. 485-498, 2016.

[39] G. Q. Li, J. Zhao, V. Murray, C. Song, L. C. Zhang, Gap analysis on open data interconnectivity for disaster risk research, *Geo-Spatial Information Science*, Vol. 22, No. 1, pp. 45-58, January, 2019.

[40] A. Tiernan, L. Drennan, J. Nalau, E. Onyango, L. Morrissey, B. Mackey, A review of themes in disaster resilience literature and international practice since 2012, *Policy design and practice*, Vol. 2, No. 1, pp. 53-74, 2019.

[41] M. Hidalgo-Herrero, P. Rabanal,, I. Rodriguez, F. Rubio, Comparing problem solving strategies for NP-hard optimization problems, *Fundamenta Informaticae*, Vol. 124, No. 1-2, pp. 1-25, January, 2013.

[42] P. Toth, Optimization engineering techniques for the exact solution of NP-hard combinatorial optimization problems, *European journal of operational research*, Vol. 125, No. 2, pp. 222-238, September, 2000.

[43] I. M. Ali, D. Essam, K. Kasmarik, Novel binary differential evolution algorithm for knapsack problems, *Information Sciences*, Vol. 542, pp. 177-194, January, 2021.

[44] I. Aljarah, M. Mafarja, A. A. Heidari, H. Faris, S. Mirjalili, Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach, *Knowledge and Information Systems*, Vol. 62, No. 2, pp. 507-539, February, 2020.

[45] X. J. Zhao, C. Wang, J. X. Su, J. Z. Wang, Research and application based on the swarm intelligence algorithm and artificial intelligence for wind farm decision system, *Renewable energy*, Vol. 134, pp. 681-697, April, 2019.

[46] J. Z. Wang, P. Du, T. Niu, W. D. Yang, A novel hybrid system based on a new proposed algorithm—Multi-Objective Whale Optimization Algorithm for wind speed forecasting, *Applied energy*, Vol. 208, pp. 344-360, December, 2017.

[47] L. Wang, L. Peng, S. R. Wang, S. Liu, Advanced backtracking search optimization algorithm for a new joint replenishment problem under trade credit with grouping constraint, *Applied Soft Computing*, Vol. 86, Article No. 105953, January, 2020.

[48] S. S. Shekhawat, H. Sharma, S. Kumar, A. Nayyar, B. Qureshi, bSSA: Binary Salp Swarm Algorithm With Hybrid Data Transformation for Feature Selection, *IEEE Access*, Vol. 9, pp. 14867-14882, January, 2021.

[49] Y. Xue, H. K. Zhu, J. Y. Liang, A. Stowik, Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification, *Knowledge-Based Systems*, Vol. 227, Article No. 107218, September, 2021.

[50] A. D. Li, B. Xue, M. J. Zhang, Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies, *Applied Soft Computing*, Vol. 106, Article No. 107302, July, 2021.

[51] H. Mohmmadzadeh, F. S. Gharehchopogh, An efficient binary chaotic symbiotic organisms search algorithm approaches for feature selection problems, *The Journal of Supercomputing*, Vol. 77, pp. 9102-9144, August, 2021.

[52] L. F. Chen, C. T. Su, K. H. Chen, An improved particle swarm optimization for feature selection, *Intelligent Data Analysis*, Vol. 16, No. 2, pp. 167-182, March, 2012.

[53] J.-S. Pan, P. Hu, S.-C. Chu, Binary fish migration optimization for solving unit commitment, *Energy*, Vol. 226, Article No. 120329, July, 2021.

[54] R. Algin, A. F. Alkaya, M. Agaoglu, Feature selection via computational intelligence techniques, *Journal of Intelligent & Fuzzy Systems*, Vol. 39, No. 5, pp. 6205-6216, November, 2020.

[55] J.-S. Pan, A. Q. Tian, S.-C. Chu, J. B. Li, Improved binary pigeon-inspired optimization and its application for feature selection, *Applied Intelligence*, Vol. 51, No. 12, pp. 8661-8679, December, 2021.

[56] P. Hu, J.-S. Pan, S.-C. Chu, Improved Binary Grey Wolf

Optimizer and Its Application for Feature Selection, *Knowledge-Based Systems*, Vol. 195, Article No. 105746, May, 2020.

[57] C. J. Barnard, R. M. Sibly, Producers and scroungers: a general model and its application to captive flocks of house sparrows, *Animal behaviour*, Vol. 29, No. 2, pp. 543-550, May, 1981.

[58] D. Dua, C. Graff, UCI machine learning repository University of California, *Irvine, School of Information and Computer Sciences*, 2017.

[59] R. Harun, K. C. Pelangi, Y. Lasena, Penerapan Data Mining untuk Menentukan Potensi Hujan Harian dengan Menggunakan Algoritma K Nearest Neighbor (KNN), *Jurnal Manajemen Informatika dan Sistem Informasi*, Vol. 3, No. 1, pp. 8-15, January, 2020.

Biographies

Xu Yuan received his B.S.degree from Liaocheng university, Liaocheng, China, in 2021. He is currently pursuing the master degree with the Shandong University of Science and Technology, Qingdao, China. His recent research interests are Swarm Intelligence and Intelligent Computing.

Jeng-Shyang Pan received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology in 1986, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from the University of Edinburgh, U.K., in 1996. He is currently the Director of the Fujian Provincial Key Lab of Big Data Mining and Applications, and an Assistant President with the Fujian University of Technology. He is also the Professor with the Harbin Institute of Technology. He is the IET Fellow, U.K., and has been the Vice Chair of the IEEE Tainan Section. He was offered Thousand Talent Program in China in 2010.

Ai-Qing Tian received his B.S. degree from Taishan College of Science and Technology, Taian, China, in 2019. He is currently pursuing the master degree with the Shandong University of Science and Technology, Qingdao, China. His recent research interests are swarm intelligence and artificial neural networks.

Shu-Chuan Chu received the Ph.D. degree in 2004 from the School of Computer Science, Engineering and Mathematics, Flinders University of South Australia. She joined Flinders University in December 2009 after 9 years at the Cheng Shiu University, Taiwan. She is the Research Fellow in the College of Science and Engineering of Flinders University, Australia from December 2009. Currently, She is the Research Fellow with PhD advisor in the College of Computer Science and Engineering of Shandong University of Science and Technology from September 2019. Her research interests are mainly in Swarm Intelligence, Intelligent Computing and Data Mining.