ARTICLE IN PRESS

Knowledge-Based Systems xxx (xxxx) xxx

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Improved Binary Grey Wolf Optimizer and Its application for feature selection☆

Pei Hu [a,b], Jeng-Shyang Pan [a], Shu-Chuan Chu [a,c,*]

[a] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[b] School of Software, Nanyang Institute of Technology, Nanyang 473004, China
[c] College of Science and Engineering, Flinders University Sturt Rd, Bedford Park SA 5042, Australia

ARTICLE INFO

Article history:
Received 24 December 2019
Received in revised form 19 February 2020
Accepted 6 March 2020
Available online xxxx

Keywords:
Grey Wolf Optimizer
Discrete
Binary
Transfer function
Feature selection

ABSTRACT

Grey Wolf Optimizer (GWO) is a new swarm intelligence algorithm mimicking the behaviours of grey wolves. Its abilities include fast convergence, simplicity and easy realization. It has been proved its superior performance and widely used to optimize the continuous applications, such as, cluster analysis, engineering problem, training neural network and etc. However, there are still some binary problems to optimize in the real world. Since binary can only be taken from values of 0 or 1, the standard GWO is not suitable for the problems of discretization. Binary Grey Wolf Optimizer (BGWO) extends the application of the GWO algorithm and is applied to binary optimization issues. In the position updating equations of BGWO, the *a* parameter controls the values of *A* and *D*, and influences algorithmic exploration and exploitation. This paper analyses the range of values of *AD* under binary condition and proposes a new updating equation for the *a* parameter to balance the abilities of global search and local search. Transfer function is an important part of BGWO, which is essential for mapping the continuous value to binary one. This paper includes five transfer functions and focuses on improving their solution quality. Through verifying the benchmark functions, the advanced binary GWO is superior to the original BGWO in the optimality, time consumption and convergence speed. It successfully implements feature selection in the UCI datasets and acquires low classification errors with few features.

1. Introduction

In the information systems, thousands of applications have generated datasets increasingly and decision makers need to apply various analytical techniques to extract valuable information from data sources, such as statistical analysis, data mining and neural network [1–4]. There has been an increase in practical applications' features for pattern recognition and machine learning. Among these features, a number may be unrelated or redundant, but it is difficult to estimate which one is useful in the absence of prior knowledge [5–7]. In such situations, feature selection becomes great significant. Feature selection defines the process for selecting several of the most effective features from the original features to reduce the dimension of a dataset. The selected subset also requires high classification accuracy for the original dataset [8–10].

The algorithms of feature selection are usually divided into 3 categories: filter, wrapper and embedded [11–13]. The filter algorithm selects features from the dataset and then uses the selected feature subset to train the learner. The feature selection and the subsequent learner processes are independent of each other. The wrapper algorithm adopts the evaluation criterion which takes the performance of the learner to acquire feature selection. The embedded algorithm is completed in the same optimization process for feature selection and the process of training the learner.

Feature selection is a NP-hard problem, with n being the number of features and its search space being 2^n. It is impossible to thoroughly search all possible solutions in most cases. Many researchers have to introduce various approximation algorithms to acquire near-optimal solutions due to the high complexity for feature selection. Over the last decades, meta heuristic algorithm has become a useful tool to solve complex optimization problems [14–18]. It acquires the optimal solution or approximate one for the problem based on acceptable calculation space or time. As a part of meta heuristic, swarm intelligence is a kind

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2020.105746.

* Corresponding author at: College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China.
E-mail addresses: hupei@nyist.edu.cn (P. Hu), jspan@cc.kuas.edu.tw (J. Pan), jan.chu83@gmail.com (S. Chu).

https://doi.org/10.1016/j.knosys.2020.105746
0950-7051/

Please cite this article as: P. Hu, J. Pan and S. Chu, Improved Binary Grey Wolf Optimizer and Its application for feature selection, Knowledge-Based Systems (2020) 105746, https://doi.org/10.1016/j.knosys.2020.105746.

of bionic algorithm based on population. It has attracted much attention because of its simplicity and easy implementation. Particle Swarm Optimization (PSO) simulates the foraging behaviour of birds to find the optimal solution [19–26]. Ant Colony Optimization (ACO) imitates the process of ants scavenging for food to solve the optimal path [27–29]. Cat Swarm Optimization (CSO) mimics cats seeking for their prey [30–32]. Grey Wolf Optimizer (GWO) is an optimization algorithm based on the hierarchy and hunting behaviour of a wolf pack [33–35]. QUasi-Affine TRansformation Evolutionary (QUATRE) is a co-evolutionary framework for quasi-affine transformation and allows statistical and probabilistic searchings [36–39]. Meanwhile, with the development of neural network, deep learning has become a hot research field. Due to its flexibility and effectiveness, Abualigah et al. successfully use it for text feature extraction and achieve great results [40,41].

In the real world, there are many optimization fields, such as, supply chain [42–46], transportation planning [47,48], total interpretive structural model [49], economic order quantity model [50], lot-sizing [51], selective maintenance scheduling [52], integrated sustainable manufacturing [53], etc. Supply chain is one of the most major parts of service business and production. It contains material, information, funding and knowledge flows, and refers to meet customer service level while minimizing total costs. Abolfazl Gharaei et al. have made an in-depth studies in the areas of multiple suppliers, uncertain demands, trade credits and limited resources [54–56]. Transportation planning, lot-sizing, selective maintenance scheduling and integrated sustainable manufacturing are vital to improve production efficiency and save resources. For deep reviews of other solution methods, the below references are considered [57–61].

Nonetheless, many real applications are discrete problems, such as feature selection, 0–1 knapsack problem and travelling salesman problem. In order to handle the problems by swarm intelligence, continuous values have to be transformed into binary ones, so the algorithms need corresponding binary versions. Binary Grey Wolf Optimizer (BGWO) is first proposed by E. Emary et al. [62]. AL-TASHI et al. and Panwar et al. utilizes BGWO to solve the problems of feature selection and large scale unit commitment [63,64]. Srikanth et al. introduces quantum and GWO (QI-BGWO) to solve unit commitment problem [65].

This paper focuses on balancing the abilities of exploration and exploitation for binary GWO. At the early stage, the algorithm executes exploration and searches for new space as far as possible. At the late stage, the algorithm performs exploitation and searches carefully in the potential space to find the optimal solution. Consequently, the algorithm has fast position switching at first, and with the evolution of the algorithm, it gradually slows down the speed of position conversion. Transfer function is a key measure for controlling global search and local search, and its mathematical properties also have critical effects on the performance of binary GWO. It is necessary to introduce new transfer function to improve the solution quality. Transfer function maps the value of AD to [0,1], thus the range of AD influences the achievement of the algorithm, so the a parameter should be paid special attention. The main contributions of this paper are concluded as follows:

(1) It analyses the range values of the parameters under the Binary Grey Wolf Optimizer.

(2) It proposes a new updating equation for a parameter.

(3) It introduces new transfer functions and improves them based on the results of (1).

(4) Through the benchmark functions, it testifies the performance of the improved Binary Grey Wolf Optimizer algorithms.

(5) It implements feature selection based on improved Binary Grey Wolf Optimizer algorithms.

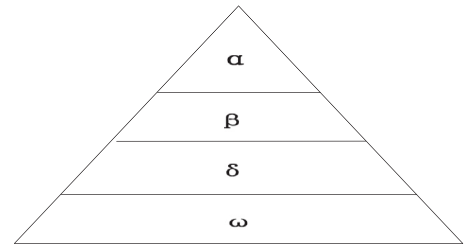

Fig. 1. The hierarchy of the grey wolf pack.

The rest of the paper is organized as follows. Section 2 describes GWO and binary GWO. Section 3 investigates the restriction conditions for the parameters of binary GWO. It proposes the new updating equation for a parameter and improves transfer functions based on exploration and exploitation. Section 4 illustrates the performance of solution quality, convergence and computational time by benchmark functions. Section 5 concentrates on feature selection in the UCI data sets by the proposed methods. Section 6 depicts the main works of the paper and gives some suggestions for further works.

2. Related works

Grey Wolf Optimizer is a new population-based swarm intelligence algorithm introduced by Mirjalili et al. In this section, it will present the basic theories of GWO and binary GWO.

2.1. The standard Grey Wolf Optimizer

As shown in Fig. 1, the grey wolf pack has a very strict social hierarchy similar to the pyramid. The first three leaders are called alpha (α), beta (β) and delta (δ). The omega (ω) wolves are at the bottom. The alpha heads the wolf pack to hunt prey and the omegas receive the leaderships of the three leaders.

Grey Wolf Optimizer (GWO) utilizes the mechanisms of hierarchy, searching and hunting for prey, much like the wolf pack would do in nature. Every wolf represents a potential candidate solution and the prey is the optimum. α, β and δ are the first three optimal solutions. The remaining candidates are collectively referred to as ω. Based on the locations of α, β and δ, the omegas update their positions. A wolf (i) calculates its distance from the three optimal solutions by Eqs. (1)–(4) and then uses Eq. (5) to update its position.

$$\vec{A} = 2\vec{a} \cdot \vec{r_1} - \vec{a} \tag{1}$$

$$\vec{B} = 2\vec{r_2} \tag{2}$$

$$\vec{D_\alpha} = |\vec{B} \cdot \vec{X_\alpha} - \vec{X_i}|, \vec{D_\beta} = |\vec{B} \cdot \vec{X_\beta} - \vec{X_i}|, \vec{D_\delta} = |\vec{B} \cdot \vec{X_\delta} - \vec{X_i}| \tag{3}$$

$$\vec{X_1} = \vec{X_\alpha} - \vec{A} \cdot \vec{D_\alpha}, \vec{X_2} = \vec{X_\beta} - \vec{A} \cdot \vec{D_\beta}, \vec{X_3} = \vec{X_\delta} - \vec{A} \cdot \vec{D_\delta} \tag{4}$$

$$\vec{X_i}(nt) = \frac{\vec{X_1} + \vec{X_2} + \vec{X_3}}{3} \tag{5}$$

where $\vec{X_i}$, $\vec{X_\alpha}$, $\vec{X_\beta}$ and $\vec{X_\delta}$ represent the position vectors of i, α, β and δ; nt denotes the next iteration; $\vec{D_\alpha}$, $\vec{D_\beta}$ and $\vec{D_\delta}$ respectively mean the distance vectors between α, β, δ and i. Both $\vec{r_1}$ and $\vec{r_2}$ are random vectors between [0, 1]; $\vec{A}$ and $\vec{B}$ are two coefficient vectors. a is calculated as follows:

$$a = 2 - 2 * it/MAX\_IT \tag{6}$$

Please cite this article as: P. Hu, J. Pan and S. Chu, Improved Binary Grey Wolf Optimizer and Its application for feature selection, Knowledge-Based Systems (2020) 105746, https://doi.org/10.1016/j.knosys.2020.105746.

where *it* is the present number of iteration and *MAX_IT* indicates the max number of iteration. The pseudo code of GWO is presented in Algorithm 1.

Algorithm 1 GWO

Initialize the related parameters of GWO
Randomly generate the positions of the wolves
Compute the fitness of each wolf
Find X_α, X_β and X_δ
for $i = 1 : MAX\_IT$ **do**
  Update a, A and B by Eqs. (1), (2) and (6)
  Calculate the position of each wolf by Eqs. (3)–(5)
  Compute the fitness of each wolf
  Update X_α, X_β and X_δ
end for
Output X_α

2.2. Binary Grey Wolf Optimizer

In the GWO, the positions are situated at any points in the continuous space. Consequently, the updating equations can be easily implemented. In the Binary Grey Wolf Optimizer (BGWO), the search space is considered a hypercube and the positions of wolves are only located in the values of 0 or 1. The wolves shift their locations to be closer of the hypercube or further away from it through altering some numbers, therefore it can-not be updated by using the same equations.

BGWO adopts the same strategy to acquire the values of α, β and δ, and uses Eq. (3) to calculate $\overrightarrow{D_\alpha}$, $\overrightarrow{D_\beta}$ and $\overrightarrow{D_\delta}$. Then it obtains s_1, s_2 and s_3 by using sigmoid function (called S_1), as the follows.

$$s_1^d = 1/(1 + e^{-10(A^d \cdot D_\alpha^d - 0.5)}) \tag{7}$$

$$s_2^d = 1/(1 + e^{-10(A^d \cdot D_\beta^d - 0.5)}) \tag{8}$$

$$s_3^d = 1/(1 + e^{-10(A^d \cdot D_\delta^d - 0.5)}) \tag{9}$$

where d is the dth dimension of an agent (wolf).

The values of $bstep_1$, $bstep_2$ and $bstep_3$ are computed using Eqs. (10)–(12). After this step, the result will be a binary value and no longer a continuous one. It then uses the transfer function to switch, as seen in Eqs. (7)–(9). The values of 0 and 1 are required for comparison with random numbers.

$$bstep_1^d = \begin{cases} 1 & if(s_1^d \geq randn) \\ 0 & else \end{cases} \tag{10}$$

$$bstep_2^d = \begin{cases} 1 & if(s_2^d \geq randn) \\ 0 & else \end{cases} \tag{11}$$

$$bstep_3^d = \begin{cases} 1 & if(s_3^d \geq randn) \\ 0 & else \end{cases} \tag{12}$$

where *randn* is a random number between [0, 1]. $bstep_1$, $bstep_2$ and $bstep_3$ are the distances that i will move relative to α, β and δ. Next, X_1, X_2 and X_3 are calculated by the following equations.

$$X_1^d = \begin{cases} 1 & if(X_\alpha^d + bstep_1^d \geq 1) \\ 0 & else \end{cases} \tag{13}$$

$$X_2^d = \begin{cases} 1 & if(X_\beta^d + bstep_2^d \geq 1) \\ 0 & else \end{cases} \tag{14}$$

$$X_3^d = \begin{cases} 1 & if(X_\delta^d + bstep_3^d \geq 1) \\ 0 & else \end{cases} \tag{15}$$

In the end, it adopts a simple stochastic crossover to update the position of X_i in the next iteration, as shown in Eq. (16). Algorithm 2 is the pseudo code of BGWO.

$$X_i^d(nt) = \begin{cases} X_1^d & if(rand < 1/3) \\ X_2^d & elseif(1/3 \leq rand < 2/3) \\ X_3^d & else \end{cases} \tag{16}$$

Algorithm 2 BGWO

Initialize the related parameters of BGWO
Randomly generate the positions of the wolves
Compute the fitness of each wolf
Find X_α, X_β and X_δ
for $i = 1 : MAX\_IT$ **do**
  Update a, A and B by Eqs. (1), (2) (6)
  Calculate the position of each wolf by Eqs. (7)–(16)
  Compute the fitness of each wolf
  Update X_α, X_β and X_δ
end for
Output X_α

3. Analysis and advancement for Binary Grey Wolf Optimizer

In this section, the range of values for AD is analysed based on binary GWO and new transfer functions are introduced to substitute for the sigmoid function. Then a new equation for a parameter and improved transfer functions are proposed to increase the solution quality.

3.1. Mathematical analysis

The positions of wolves locate to whatever points there are in the space of GWO. But in the Binary Grey Wolf Optimizer, their positions only locate in the values of 0 and 1, so binary GWO needs to meet the restriction and has its own structure. In order to simply depict the mathematical model, only one dimension will be considered, with X_i^d being represented by X_i.

From Eq. (1), it is known that $A = (2ar_1 - a)$ and $D_\alpha = |2r_2X_\alpha - X_i|$, so $AD_\alpha = (2ar_1 - a)|2r_2X_\alpha - X_i|$. Since X_α and X_i only take values from {0,1}, there are four combinations of their values and the value of AD_α is computed as follows:

(1) if $X_\alpha = 0$ and $X_i = 0$
$AD_\alpha = (2ar_1 - a)|2r_2X_\alpha - X_i| = 0$
(2) if $X_\alpha = 0$ and $X_i = 1$
$AD_\alpha = (2ar_1 - a)|2r_2X_\alpha - X_i| = a(2r_1 - 1)$
Since $r_1 \in [0, 1]$, then $(2r_1 - 1) \in [-1, 1]$.
Because $a \in [0, 2]$, thereby $a(2r_1 - 1) \in [-2, 2]$, namely $AD_\alpha \in [-2, 2]$.
(3) if $X_\alpha = 1$ and $X_i = 0$
$AD_\alpha = (2ar_1 - a)|2r_2X_\alpha - X_i| = a(2r_1 - 1)2r_2 = 2ar_2(2r_1 - 1)$
Since $r_1 \in [0, 1]$, $r_2 \in [0, 1]$, $a \in [0, 2]$, then $2ar_2 \in [0, 4]$ and $(2r_1 - 1) \in [-1, 1]$.
So, $AD_\alpha \in [-4, 4]$.
(4) if $X_\alpha = 1$ and $X_i = 1$
$AD_\alpha = (2ar_1 - a)|2r_2X_\alpha - X_i| = (2ar_1 - a)|2r_2 - 1|$
Since $r_1 \in [0, 1]$, $r_2 \in [0, 1]$, $a \in [0, 2]$, then $(2ar_1 - a) \in [-2, 2]$ and $|2r_2 - 1| \in [0, 1]$.
Hence, $AD_\alpha \in [-2, 2]$.

From the above analysis, it can be concluded that $AD_\alpha \in [-4, 4]$. Similarly, AD_β and AD_δ all $\in [-4, 4]$, that is $AD \in [-4, 4]$ as well. In the following sections, the achieved results will be used for further discussion.

Please cite this article as: P. Hu, J. Pan and S. Chu, Improved Binary Grey Wolf Optimizer and Its application for feature selection, Knowledge-Based Systems (2020) 105746, https://doi.org/10.1016/j.knosys.2020.105746.

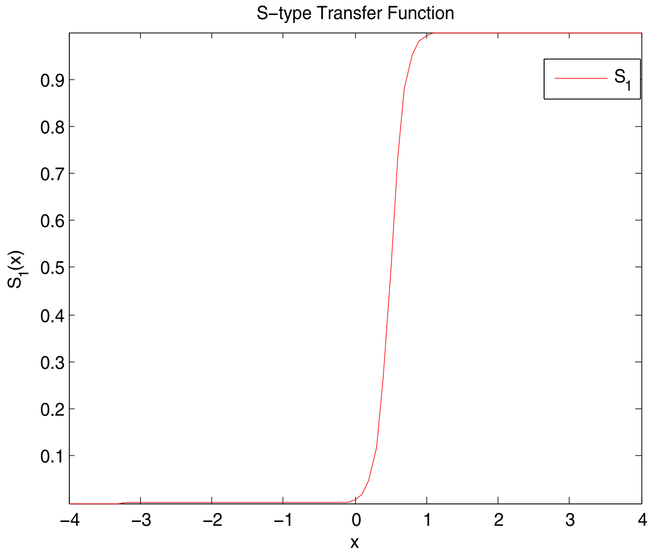

Fig. 2. The curve of S_1.

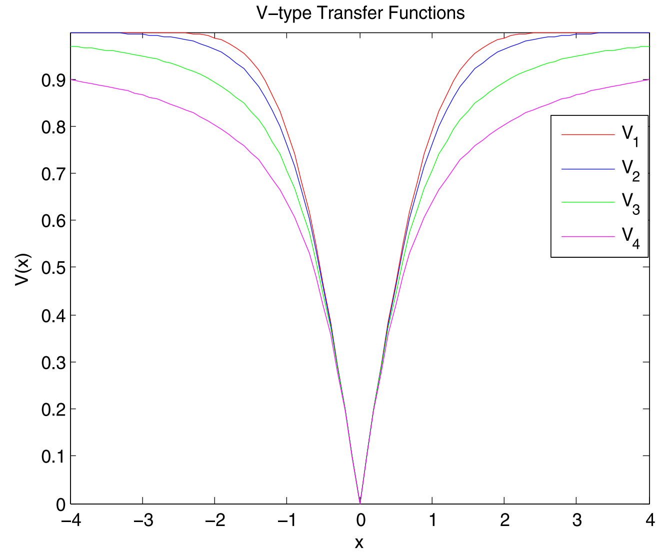

Fig. 3. The curves of V-type transfer functions.

Table 1
The details of V-type transfer functions.

Name	Transfer function
$V_1(x)$	$\left\lvert\frac{\sqrt{2}}{\pi}\int_{-0}^{(\sqrt{\pi}/2)x} e^{t^2}\,dt\right\rvert$
$V_2(x)$	$\lvert tanh(x)\rvert$
$V_3(x)$	$\lvert x/\sqrt{1+x^2}\rvert$
$V_4(x)$	$\lvert\frac{2}{\pi}arctan(\frac{2}{\pi}x)\rvert$

Table 2
The details of the derivative functions.

Transfer function	Derivative function
$V_1(x)$	$D_1(x)=\begin{cases}\frac{1}{\sqrt{2\pi}}e^{-\frac{\pi}{4}x^2} & x\geq 0\\ -\frac{1}{\sqrt{2\pi}}e^{-\frac{\pi}{4}x^2} & x<0\end{cases}$
$V_2(x)$	$D_2(x)=\begin{cases}1-tanh^2(x) & x\geq 0\\ -1/(1+x^2)^{\frac{2}{3}} & x<0\end{cases}$
$V_3(x)$	$D_3(x)=\begin{cases}1/(1+x^2)^{\frac{2}{3}} & x\geq 0\\ -1/(1+x^2)^{\frac{2}{3}} & x<0\end{cases}$
$V_4(x)$	$D_4(x)=\begin{cases}1/(1+(\frac{\pi}{2}x)^2) & x\geq 0\\ -1/(1+(\frac{\pi}{2}x)^2) & x<0\end{cases}$
$S_1(x)$	$D_S(x)=\frac{10e^{10(x-0.5)}}{(e^{10(x-0.5)}+1)^2}$

3.2. New transfer functions

Transfer function is the simplest method in the binary GWO, which maps the continuous values to [0,1] and then discretizes them to 0 and 1 according to the probability. It keeps the structure and other operations of the GWO, and makes the positions of population move in a binary space.

From Section 2.2, BGWO is known to use S_1 as the transfer function and Fig. 2 shows the curve of S_1. The value of $S_1(x)$ increases with the value of x. For example, if $x=0.5$, $S_1(x)=0.5$; when x equals 0.5, BGWO has the probability of 50% becoming 0; if $x<0.5$, it has a large probability becoming 0; if $x>0.5$, it has a small probability becoming 0.

In this section, four transfer functions are introduced, [66] called V-shaped ones. Table 1 shows the details and Fig. 3 displays the curves of the functions.

It can be seen from Fig. 3 that the values of the new transfer functions are 0 when x equals 0 and approach 1 when the absolute value of x gets closer to infinity. So, if x equals 0, the algorithm has the high possibility to acquire 0. If the absolute value of x tends to infinity, it has the high possibility to gain 1.

3.3. New updating equation for a parameter based on exploration and exploitation

In this section, the values of AD are continued to analyse from the results of Section 3.1 and the new updating equation for a is proposed based on exploration and exploitation. It is previously known that the value of position can only be selected from 0 or 1 in the binary algorithm and the updating position represents shifting between 0 and 1 in the discrete binary space. The conversion is completed by the values of AD, so several equations require modification when designing the Binary Grey Wolf Optimizer.

At the early phase of the algorithm, the position needs to be able to rapidly switch between 0 and 1 to search more space. While at the late phase, the position switching speed is slower, to increase the ability when searching for the optimal solution. The slope of the functional curve represents numerical acceleration. Table 2 is the details of the derivative functions corresponding to the five transfer functions and Fig. 4 shows the curves of the derivative functions.

Fig. 4 implies that the larger the absolute value of AD is, the smaller the value of $D(x)$ is, namely, the smaller the slope is; similarly, the slope increases as the absolute value of AD decreases. Because the slope indicates the switching speed of position, so if the absolute value of AD is large, the position of X_i changes slowly, and if the absolute value of AD is small, the position of X_i alters rapidly. It can be seen from Section 3.1 that the absolute value of AD is large only when a parameter is large. In other words, the algorithm is in exploration and the position should change fast contradicting the slow change of X_i.

From Eqs. (1) and (3), a parameter influences the values of A and D and further affects the values of AD. In order to make the algorithm with fast position switching in exploration and great optimization ability in exploitation, the updating equation for a

Please cite this article as: P. Hu, J. Pan and S. Chu, Improved Binary Grey Wolf Optimizer and Its application for feature selection, Knowledge-Based Systems (2020) 105746, https://doi.org/10.1016/j.knosys.2020.105746.

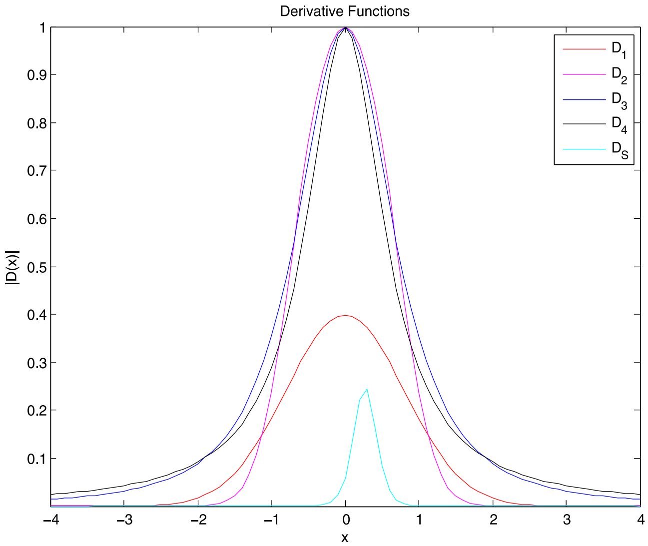

Fig. 4. The curves of the derivative functions.

is redefined as follows:

$$a = 2 * it/MAX\_IT \quad (17)$$

From Eq. (17), a linearly increases from 0 to 2. In the initial stage of the algorithm, a is a small numerical value. That means, the value of AD is small and the binary GWO has a large probability of position switching. With the evolution of the algorithm, a becomes large and AD has a high probability of obtaining the maximum value. Hence, the binary GWO has a small probability of changing position. This is an exceptional balance between the exploration and exploitation.

3.4. Improved transfer functions

Transfer function is a key part of the binary GWO and is an easy method to control exploration and exploitation of the algorithm. In Section 3.2, four new V-type transfer functions are introduced and this section will further improve them based on the results of Section 3.1.

Firstly, from Eqs. (10) and (12) known that the BGWO compares the value of transfer function with a random number between [0, 1], so the value should be at [0, 1]. Secondly, the function provides a high probability for position switching when the absolute value of AD is large and a small probability for position switching when the absolute value of AD is small.

The transfer function controls the rate at which 0 and 1 are switched. The performance of the binary GWO may degrade through using an unsuitable transfer function. From Fig. 3 it is seen that $V(x)$ is close to 1 when the value of x is infinite. However, in the binary GWO, the value of x is at $[-4, 4]$. The max values of $V_1(x)$, $V_2(x)$, $V_3(x)$ and $V_4(x)$ are 1.0000, 0.9993, 0.9701 and 0.8955. The original purpose is to have a high probability of becoming 1 when the value is large. Now even if x reaches the maximum value, $V_2(x)$, $V_3(x)$ and $V_4(x)$ have the probabilities of 0.07%, 2.99% and 10.45% that they do not reach 1 and this contradicts the original goal. In order to avoid the situation, the transfer functions need to be stretched for the special case of binary GWO and also need to ensure that the values of transfer functions are at [0,1]. The improved transfer functions are shown in Table 3.

where $tanh(4)$, $4/\sqrt{17}$ and $arctan(2\pi)$ are the max values of $V_2(x)$, $V_3(x)$ and $V_4(x)$ in the binary GWO.

Table 3
The details of the improved transfer functions.

Name	Transfer function
$V_{2a}(x)$	$\lvert tanh(x)\rvert/tanh(4)$
$V_{3a}(x)$	$\lvert\sqrt{17} * x/(4 * \sqrt{1+x^2})\rvert$
$V_{4a}(x)$	$\lvert arctan(\frac{2}{\pi}x)/arctan(2\pi)\rvert$

Table 4
Unimodal benchmark functions.

Name	Function	Space	D_{im}	f_{min}
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	$[-100, 100]$	30	0
Schwefel's function 2.21	$f_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	$[-10, 10]$	30	0
Schwefel's function 1.2	$f_3(x) = \sum_{i=1}^{n}(\sum_{j-1}^{i} x_j)^2$	$[-100, 100]$	30	0
Schwefel's function 2.22	$f_4(x) = max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	$[-100, 100]$	30	0
Rosenbrock	$f_5(x) = \sum_{i=1}^{n-1}[100 (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]$	30	0
Step	$f_6(x) = \sum_{i=1}^{n}([x_i + 0.5])^2$	$[-100, 100]$	30	0
Dejong's noisy	$f_7(x) = \sum_{i=1}^{n} ix_i^4 + random[0, 1)$	$[-1.28, 1.28]$	30	0

4. Experimental results and analysis

In this section, the general abilities of exploration and exploitation of the proposed methods are examined by 29 benchmark functions. Tables 4 to 7 describe the details of unimodal, multimodal, fixed-dimension and composite functions, respectively. *Space* denotes the boundary of search space; D_{im} means the dimension of function and f_{min} represents the optimum.

Unimodal function only has a global optimal solution and no local trap, so it can verify the convergence speed of the algorithm. While multimodal function has exponential local optimal solutions, it can judge whether the algorithm avoids falling into local trap. Composite multimodal function has a complex structure with many random global optimal solutions and several random deep local optimums. So, through the benchmark functions, the algorithm may be tested from different aspects.

For verifying the results, BGWO is compared with the V-type transfer functions, advanced transfer functions and new updating equation for a. They run 30 times and 500 iterations on each benchmark function. The population has 30 individuals. Table 8 shows the details of the compared methods.

4.1. Experimental results

To facilitate reading, the experimental data is rounded to four decimal places. Tables 9 and 10 show the average (AVG) and standard deviation (STD) of each function in the original updating equation for a parameter.

Table 11 shows the statistical results of each function in new equation for a. *Winner* indicates which algorithm performs the best in the benchmark function and *all* means that all compared algorithms have the same performance.

Table 5
Multimodal benchmark functions.

Name	Function	Space	D_{im}	f_{min}
Schwefel	$f_8(x) = \sum_{i=1}^{n} -x_i sin(\sqrt{\lvert x_i \rvert})$	[−500,500]	30	−12 569
Rastringin	$f_9(x) = \sum_{i=1}^{n} [x_i^2 - 10cos(2\pi x_i) + 10]$	[−5.12,5.12]	30	0
Ackley	$f_{10}(x) = -20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) -exp(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)) + 20 + e$	[−32,32]	30	0
Griewank	$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}) + 1$	[−600,600]	30	0
Generalized penalized 1	$f_{12}(x) = \frac{\pi}{n}\{10sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2 [1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4) y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	[−50,50]	30	0
Generalized penalized 2	$f_{13}(x) = 0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2 [1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	[−50,50]	30	0

Table 6
Fixed-dimension benchmark functions.

Name	Function	Space	D_{im}	f_{min}
Fifth of Dejong	$f_{14}(x) = (\frac{1}{500}\sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	[−65, 65]	2	1
Kowalik	$f_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	[−5, 5]	4	0.00030
Six-hump camel back	$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	[−5, 5]	2	−1.0316
Branins	$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})cosx_1 + 10$	[−5, 5]	2	0.398
Goldstein–Price	$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	[−2, 2]	2	3
Hartman 1	$f_{19}(x) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	[1, 3]	3	−3.86
Hartman 2	$f_{20}(x) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	[0, 1]	6	−3.32
Shekel 1	$f_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]	4	−10.1532
Shekel 2	$f_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]	4	−10.4028
Shekel 3	$f_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]	4	−10.5363

Please cite this article as: P. Hu, J. Pan and S. Chu, Improved Binary Grey Wolf Optimizer and Its application for feature selection, Knowledge-Based Systems (2020) 105746, https://doi.org/10.1016/j.knosys.2020.105746.

Table 7
Composite benchmark functions.

Name	Function	Space	D_{im}	f_{min}
CF1	f_{24} $f_1, f_2, f_3, \ldots, f_{10} = Sphere\ \ Function$ $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [5/100, 5/100, 5/100, \ldots, 5/100]$	[−5, 5]	30	0
CF2	f_{25} $f_1, f_2, f_3, \ldots, f_{10} = Griewank's\ \ Function$ $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [5/100, 5/100, 5/100, \ldots, 5/100]$	[−5, 5]	30	0
CF3	f_{26} $f_1, f_2, f_3, \ldots, f_{10} = Griewank's\ \ Function$ $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [1, 1, 1, \ldots, 1]$	[−5, 5]	30	0
CF4	f_{27} $f_1, f_2 = Ackley'sFunction, f_3, f_4 = Rastrigin's\ \ Function,$ $f_5, f_6 = WeierstrassFunction, f_7, f_8 = Griewank's\ \ Function$ $f_9, f_{10} = Sphere\ \ Function$ $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] =$ $[5/32, 5/32, 1, 1, 5/0.5, 5/0.5, 5/100, 5/100, 5/100, 5/100]$	[−5, 5]	30	0
CF5	f_{28} $f_1, f_2 = Rastrigin's\ \ Function, f_3, f_4 = Weierstrass\ \ Function,$ $f_5, f_6 = Griewank'sFunction, f_7, f_8 = Ackley's\ \ Function$ $f_9, f_{10} = Sphere\ \ Function$ $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] =$ $[1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$	[−5, 5]	30	0
CF6	f_{29} $f_1, f_2 = Rastrigin's\ \ Function, f_3, f_4 = Weierstrass\ \ Function,$ $f_5, f_6 = Griewank's\ \ Function, f_7, f_8 = Ackley's\ \ Function$ $f_9, f_{10} = Sphere\ \ Function$ $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [0.1 * 1/5, 0.2 * 1/5, 0.3 * 5/0.5, 0.4 * 5/0.5,$ $0.5 * 5/100, 0.6 * 5/100, 0.7 * 5/32, 0.8 * 5/32, 0.9 * 5/100, 1 * 5/100]$	[−5, 5]	30	0

Table 8
The details of the compared methods.

Algorithm	Updating equation	Transfer function
BGWO	Eq. (6)	S_1
BGWO_V1	Eq. (6)	V_1
BGWO_V2	Eq. (6)	V_2
BGWO_V3	Eq. (6)	V_3
BGWO_V4	Eq. (6)	V_4
BGWO_V2a	Eq. (6)	V_{2a}
BGWO_V3a	Eq. (6)	V_{3a}
BGWO_V4a	Eq. (6)	V_{4a}
ABGWO	Eq. (17)	S_1
ABGWO_V1	Eq. (17)	V_1
ABGWO_V2a	Eq. (17)	V_{2a}
ABGWO_V3a	Eq. (17)	V_{3a}
ABGWO_V4a	Eq. (17)	V_{4a}

4.2. Experimental analysis

This section analyses the performance of the compared algorithms from solution quality, time consumption and convergence speed.

4.2.1. Experiments to analyse the improved transfer functions

In Table 9, the cell is blue which indicates that the method performs well in the benchmark function. From Table 9, it appears that BGWO performs the best in the 29 benchmark functions and the newly introduced V-type transfer functions achieve poorly. However, Table 10 shows that the improved V-type transfer functions have excellent performance, especially BGWO_V3a which illustrates an outstanding achievement. Except for f_5, f_8, f_{13}, f_{21}, f_{23} and f_{24}, BGWO_V3a is better than BGWO. In Table 10, if the cell is blue, the improved transfer function performs better than the original one, and if the cell is red, it is worse than the original transfer function. There are 10, 16 and 8 in BGWO_V2a, BGWO_V3a and BGWO_V4a, respectively, which are superior to BGWO_V2, BGWO_V3 and BGWO_V4, and there are 7, 5 and 10 are inferior to BGWO_V2, BGWO_V3 and BGWO_V4. This shows that stretching transfer functions to [0,1] is conducive to improving the performance of the Grey Wolf Optimizer under the condition of binary. Each transfer function can reach the maximum value, which increases the possibility of the value of the algorithm becoming 1 and reduces the probability of it becoming 0. Compared with BGWO_V2, BGWO_V3 and BGWO_V4, BGWO_V2a, BGWO_V3a and BGWO_V4a have greatly improved their performance in the unimodal, multimodal and composite functions, while BGWO_V4a has degraded its performance in the unimodal and multimodal functions.

4.2.2. Experiments to analyse the new updating equation for a parameter

Table 11 shows the results of the improved transfer functions in the new updating equation for *a* parameter. It can be concluded that ABGWO, ABGWO_V1, ABGWO_V2a, ABGWO_V3a, and ABGWO_V4a obtain better performance than BGWO_V1, BGWO_V2a, BGWO_V3a, and BGWO_V4a in 16, 16, 15, 10 and

Please cite this article as: P. Hu, J. Pan and S. Chu, Improved Binary Grey Wolf Optimizer and Its application for feature selection, Knowledge-Based Systems (2020) 105746, https://doi.org/10.1016/j.knosys.2020.105746.

Table 9
The statistical results of the transfer functions in the original updating equation for a parameter.

Function	BGWO		BGWO_V1		BGWO_V2		BGWO_V3		BGWO_V4	
	AVG	STD	AVG	STD	AVG	STD	AVG	STD	AVG	STD
f_1	5.0333	1.1885	6.4333	1.1651	6.4333	1.3047	6.3	1.0875	6.2	1.2704
f_2	5.3333	1.5388	6.1667	1.2617	6.5333	1.1958	6.5	1.1371	6	1.0807
f_3	314.9333	145.021	617.2	202.8035	617.2333	211.3361	514.9333	220.6765	528.9667	198.4604
f_4	1	0	1	0	1	0	1	0	1	0
f_5	0	0	6.7	36.6974	26.8	69.4949	36.9333	76.9442	57.1333	73.5774
f_6	18.1	2.4155	20.3	2.5515	20.8333	2.5909	20.0333	2.2854	19.5667	1.9046
f_7	70.4667	24.0742	107.5667	19.7234	110.5334	21.8375	111.6334	18.1155	97.3667	22.2098
f_8	−25.2441	0	−25.2441	0	−25.2441	0	−25.2161	0.1536	−25.188	0
f_9	4.8333	1.1472	5.9	1.0289	6.6	1.07	6.1333	1.383	5.7667	1.2134
f_{10}	1.6176	0.1671	1.7571	0.1566	1.7298	0.2018	1.7448	0.1467	1.6961	0.1655
f_{11}	0.2465	0.0591	0.2983	0.0537	0.3056	0.0556	0.2886	0.0666	0.2844	0.0687
f_{12}	2.6872	0.2972	3.0218	0.2513	2.9959	0.3408	2.9304	0.2674	2.8704	0.2159
f_{13}	0	0	0	0	0	0	0.0033	0.0183	0.0167	0
f_{14}	12.6705	0	12.6705	0	12.6705	0	12.6705	0	12.6705	0
f_{15}	0.1484	0	0.1484	0	0.1484	0	0.1484	0	0.1484	0
f_{16}	0	0	0	0	0	0	0	0	0	0
f_{17}	27.7029	0	27.7029	0	27.7029	0	27.7029	0	27.7029	0
f_{18}	600	0	600	0	600	0	600	0	600	0
f_{19}	−0.3348	0	−0.3348	0	−0.3348	0	−0.3348	0	−0.3337	0.0063
f_{20}	−0.1507	0.0301	−0.1474	0.0373	−0.1399	0.0428	−0.1445	0.0375	−0.1506	0.0278
f_{21}	−5.0552	0	−5.0552	0	−5.0552	0	−5.0552	0	−5.0552	0
f_{22}	−5.0877	0	−5.0877	0	−5.0877	0	−5.0877	0	−5.0877	0
f_{23}	−5.1285	0	−5.1285	0	−5.1285	0	−5.1285	0	−5.1285	0
f_{24}	868.5606	6.8763	883.5155	8.5974	877.8122	5.6478	879.9331	7.6016	875.1495	7.4184
f_{25}	925.1224	4.9643	930.5019	4.3555	931.9867	4.7292	930.3518	4.5725	930.4521	4.1204
f_{26}	1148.0528	32.9742	1184.4614	26.3292	1183.2145	28.8154	1178.2798	30.5435	1170.2699	32.0997
f_{27}	1014.0455	9.0529	1026.4556	6.6383	1025.2199	7.3639	1027.095	6.1304	1021.3442	5.9148
f_{28}	1067.0443	8.0363	1080.2461	7.9196	1079.8436	9.4704	1081.0956	9.7381	1074.8607	7.241
f_{29}	925.8005	2.8173	931.0472	2.8257	930.349	4.1279	929.1108	3.0815	928.6792	3.3902

12 benchmark functions, and there are 1, 4, 5, 10, and 6 worse than BGWO_V1, BGWO_V2a, BGWO_V3a, and BGWO_V4a. It is obvious that the new equation is superior to the original equation and improves the performance in the composite functions. The solution results of ABGWO_V2a and ABGWO_V3a have been greatly improved, especially that the performance of ABGWO is not worse than BGWO in all benchmark functions. It evidences that the new updating equation for a parameter improves the exploration and exploration of the Binary Grey Wolf Optimizer. There is very small of the absolute value of AD in the early period of binary GWO, and its absolute value is large in the late period. To acquire the ideal solution, swarm intelligence algorithm needs to search more unknown space at first and gradually carefully search in the known space. It can be seen from Fig. 3 that the smaller the absolute value of AD, the faster the change. The position switching decreases as the absolute value increases, which meets the requirements of swarm intelligence algorithm. From Tables 9 to 11, the compared algorithms almost have the same solutions in the fixed-dimension functions. The reason is that the functions have small dimensions and simple structures and the new equation has more advantages in high dimension and complex structure.

4.2.3. Experiments to analyse the optimality

In order to judge whether the experimental results are statistically significant, the independent t-test method is used to make a comparison between BGWO and ABGWO, ABGWO_V1, ABGWO_V2a, ABGWO_V3a, and ABGWO_V4a. The values more than 0.05 prove that the performance of BGWO worse than others'. Table 12 shows the results. "+" appears that the compared algorithm is superior to BGWO, and "−" indicates that the algorithm is inferior to BGWO. "=" implies that the performance of the algorithms is consistent.

According to the results of Table 12, they outperform BGWO in 15, 13, 13, 14 and 13 test functions, respectively, and they have the same performance with BGWO in 14, 13, 13, 12 and 12 functions. While they have only 3, 3, 3, 4 functions that the performance is inferior to BGWO. So the proposed methods are better than BGWO in comparison of the optimality. Since the value of position is only 0 or 1, they have certain gaps between the final solutions of the benchmark functions and the theoretical optimal solutions, especially in the composite multimodal functions.

4.2.4. Experiments to analyse the time consumption and convergence

Table 13 presents the average computational time (second) of the compared algorithms. The experiments are implemented on the same computer to fair comparison. It can be seen that except for ABGWO_V3a, the compared algorithms almost have the same time consumption on the benchmark functions. This is due to the fact that the proposed algorithms only substitute the transfer function and modify the updating equation of a parameter, so the time complexity of the compared algorithms is $O(T*N*(D*CT+fobj))$, where T is the number of iteration, N represents the population size, D indicates the dimension, CT implies the time consumption of transfer function and $fobj$ means the computational time of fitness function.

ABGWO_V3a has a large time complexity. This is because erf is a Gauss error function, not an elementary function, and has a high operational time. Due to f_{14}–f_{23} with few dimensions, the algorithms have small running time. It also confirms that the dimension is an important indicator affecting the running time of the algorithm. While the algorithms have large service time on f_{24}–f_{29}. For the reason that the composite function is composed of unimodal and multimodal functions, and requires more time to run.

Rate of convergence is a significant criterion for evaluating the performance of swarm intelligence algorithm. The following equation is used to compare the rate of convergence.

$$Ratio(comp) = Iter(comp)/Iter(Opti) \tag{18}$$

where $Iter(Opti)$ indicates the number of iteration when BGWO reaches the optimum $Opti$, $Iter(comp)$ represents the number of

Table 10
The statistical results of the improved transfer functions in the original updating equation for a parameter.

Function	BGWO		BGWO_V2a		BGWO_V3a		BGWO_V4a		Winner
	AVG	STD	AVG	STD	AVG	STD	AVG	STD	
f_1	5.0333	1.1885	6.7	1.2635	3.0667	1.3374	6.2	1.2704	*BGWO_V3a*
f_2	5.3333	1.5388	6.2667	1.0483	3.4333	0.9714	6.2667	1.0807	*BGWO_V3a*
f_3	314.9333	145.021	558.0667	212.8446	79.9333	59.1695	540.2333	198.4604	*BGWO_V3a*
f_4	1	0	1	0	1	0	1	0	all
f_5	0	0	23.5667	82.622	514.1667	132.8598	13.4333	73.5774	*BGWO*
f_6	18.1	2.4155	20.1667	2.94	13.5667	2.6514	20.1	1.905	*BGWO_V3a*
f_7	70.4667	24.0742	104.9668	22.2176	47.1667	20.9977	110.3667	22.2098	*BGWO_V3a*
f_8	−25.2441	0	−25.2441	0	−23.0283	1.0478	−25.2441	0	*BGWO* *BGWO_V2a* *BGWO_V4a*
f_9	4.8333	1.1472	6.2667	1.0807	2.9667	1.129	6.1	1.2134	*BGWO_V3a*
f_{10}	1.6176	0.1671	1.8193	0.1367	1.2948	0.3571	1.7069	0.1655	*BGWO_V3a*
f_{11}	0.2465	0.0591	0.3128	0.0485	0.1374	0.0637	0.3019	0.0687	*BGWO_V3a*
f_{12}	2.6872	0.2972	2.978	0.2712	2.116	0.2587	2.9256	0.2159	*BGWO_V3a*
f_{13}	0	0	0	0	0.2633	0.1402	0	0	*BGWO* *BGWO_V2a* *BGWO_V2a*
f_{14}	12.6705	0	12.6705	0	12.6705	0	12.6705	0	all
f_{15}	0.1484	0	0.1484	0	0.1484	0	0.1484	0	all
f_{16}	0	0	0	0	0	0	0	0	all
f_{17}	27.7029	0	27.7029	0	27.7029	0	27.7029	0	all
f_{18}	600	0	600	0	600	0	600	0	all
f_{19}	−0.3348	0	−0.3348	0	−0.3348	0	−0.3348	0.0063	*BGWO* *BGWO_V2a* *BGWO_V3a*
f_{20}	−0.1507	0.0301	−0.155	0.0289	−0.1457	0.0378	−0.1566	0.0278	*BGWO_V4a*
f_{21}	−5.0552	0	−5.0552	0	−4.9161	0.7619	−5.0552	0	*BGWO* *BGWO_V2a* *BGWO_V4a*
f_{22}	−5.0877	0	−5.0877	0	−5.0877	0	−5.0877	0	all
f_{23}	−5.1285	0	−5.1285	0	−4.7105	1.2754	−5.1285	0	*BGWO* *BGWO_V2a* *BGWO_V4a*
f_{24}	868.5606	6.8763	880.8156	8.3001	870.836	6.9239	875.7851	7.4184	*BGWO*
f_{25}	925.1224	4.9643	930.1618	3.9277	922.6179	4.2119	929.7409	4.1204	*BGWO_V3a*
f_{26}	1148.0528	32.9742	1184.1633	31.3691	1092.5241	51.4146	1168.5438	32.0997	*BGWO_V3a*
f_{27}	1014.0455	9.0529	1024.0453	6.4788	987.8274	20.547	1024.5281	5.9148	*BGWO_V3a*
f_{28}	1067.0443	8.0363	1081.2865	7.4316	1048.4078	19.2377	1074.786	7.241	*BGWO_V3a*
f_{29}	925.8005	2.8173	930.7229	3.5139	924.0916	3.082	929.4294	3.3902	*BGWO_V3a*

iteration when the compared algorithm *comp* reaches *Opti*. Let $Ratio(comp) = 500$, and it means that the convergence speed of *comp* is slower than BGWO and the optimal solution of *comp* is not as good as BGWO. If $Ratio(comp)$ is less than 1, *comp* has faster convergence speed and better optimal solution than BGWO. If $Ratio(comp)$ is greater than 1 and less than 500, it implies that the convergence speed of *comp* is slower than BGWO, but the optimal solution is better than BGWO. Table 14 depicts the convergence speed of the compared algorithms.

As it can be observed that the algorithms have 4, 5, 6, 6, and 8 test functions respectively that the convergence speed is not as good as that of BGWO. ABGWO_V1, ABGWO_V2a, ABGWO_V3a and ABGWO_V4a have superior performance in the unimodal and multimodal functions. Except for ABGWO, they are all better than BGWO. The algorithms have almost the same representations in the fixed-dimension functions. It convinces that the new updating equation for a parameter greatly promotes the algorithms' rate of convergence. The new equation is able to quickly switch the positions of binary GWO and accelerate with fast convergence rate towards the global optimum over the course of iterations. The proposed transfer functions oblige wolves to switch their positions when they are moving into unpromising area of the search space.

Table 15 shows the performance of the compared algorithms on the typical features. The optimization ability under binary condition is mentioned by modifying the updating equation for a parameter. Simultaneously, the improved transfer functions also perform better by stretching the V-type transfer functions. In earlier stage, the algorithms search for the binary space as much as possible, and in the later stage, they have better optimization ability in the known space. The proposed methods have fast convergences and acquire the optimal solutions. They can avoid falling into local optimum to some extent.

5. Application for feature selection

Because there are a lot of noises and irrelevant features in data mining, it is highly difficult to deal with large data due to its dimension. Therefore, it needs to reduce the features of the dataset. Feature selection is a common technique of data mining and machine learning. Most researchers attend to the methods with high precision and low features. In this section, the wrapper method will be used to implement feature selection.

5.1. Datasets description

Twelve datasets are used to validate the performance of the BGWO and the proposed algorithms. The datasets are from UCI machine learning repository and have different numbers of attributes and instances [67], as shown in Table 16.

5.2. Simulation results

Feature selection works by selecting the most useful features from the original feature set to implement classification. In the Binary Grey Wolf Optimizer, 0 represents the unselected features and 1 means the selected features, and so the binary GWO is used to carry out feature selection.

Table 11
The statistical results of the improved transfer functions in the new updating equation for *a* parameter.

Function	ABGWO		ABGWO_V1		ABGWO_V2a		ABGWO_V3a		ABGWO_V4a		Winner
	AVG	STD	AVG	STD	AVG	STD	AVG	STD	AVG	STD	
f_1	3.1333	1.4077	3.2667	1.7407	3.2667	1.5071	3.3667	1.4967	3.0333	1.2452	*ABGWO_V4a*
f_2	3.2	1.2429	3.5	1.1963	3.467	1.383	3.3667	1.2172	3.0333	1.2994	*ABGWO_V4a*
f_3	93.0667	65.681	65.7	59.5582	87.7667	71.7994	93.2667	82.6813	88.2	56.3373	*ABGWO_V1*
f_4	1	0	1	0	1	0	1	0	1	0	all
f_5	0	0	516.7667	137.7278	506.9667	170.1824	523.3667	141.2216	455.8333	158.1654	*ABGWO*
f_6	14.3	1.8644	13.5	2.1009	13.6333	2.4598	14.1667	2.6436	13.9667	2.2702	*ABGWO_V1*
f_7	41.4334	21.2419	38.8667	20.2446	39.8334	20.4047	37.9001	20.6821	30.0001	18.2719	*ABGWO_V4a*
f_8	−25.2441	0	−22.9722	1.1725	−22.86	1.1067	−22.7478	1.0478	−22.7197	1.1268	*ABGWO*
f_9	3.3333	1.6259	3.1333	1.5916	3.1667	1.3917	3.0667	1.484	2.6333	1.1885	*ABGWO_V4a*
f_{10}	1.2178	0.335	1.2697	0.2802	1.2753	0.2846	1.2006	0.2706	1.1484	0.2865	*ABGWO_V4a*
f_{11}	0.1357	0.0594	0.146	0.0695	0.1477	0.0561	0.1268	0.0483	0.1359	0.0722	*ABGWO_V3a*
f_{12}	2.1341	0.2425	2.0689	0.2309	2.1963	0.2184	2.1446	0.2225	2.104	0.2046	*ABGWO_V1*
f_{13}	0	0	0.2767	0.1165	0.3367	0.1351	0.2833	0.1117	0.3367	0.145	*ABGWO*
f_{14}	12.6705	0	12.6705	0	12.6705	0	12.6705	0	12.6705	0	all
f_{15}	0.1484	0	0.1484	0	0.1484	0	0.1484	0	0.1484	0	all
f_{16}	0	0	0	0	0	0	0	0	0	0	all
f_{17}	27.7029	0	27.7029	0	27.7029	0	27.7029	0	27.7029	0	all
f_{18}	600	0	600	0	600	0	600	0	600	0	all
f_{19}	−0.3337	0.0063	−0.3348	0	−0.3348	0	−0.3348	0	−0.3337	0.0063	*ABGWO_V1* *ABGWO_V2a* *ABGWO_V3a*
f_{20}	−0.1566	0.0187	−0.1524	0.0208	−0.1486	0.0373	−0.1566	0.0187	−0.1367	0.0519	*ABGWO* *ABGWO_V3a*
f_{21}	−5.0552	0	−5.0552	0	−5.0552	0	−4.6379	1.2734	−5.0552	0	*ABGWO* *ABGWO_V1* *ABGWO_V2a* *ABGWO_V4a*
f_{22}	−5.0877	0	−4.9485	0.7625	−5.0877	0	−5.0877	0	−4.9485	0.7625	*ABGWO* *ABGWO_V2a* *ABGWO_V3a*
f_{23}	−5.1285	0	−5.1285	0	−4.8498	1.0604	−4.7105	1.2754	−4.5711	1.4453	*ABGWO* *ABGWO_V1a*
f_{24}	864.8001	5.1264	870.0016	7.2984	871.1298	6.5054	870.9211	7.3013	869.5212	5.7429	*ABGWO*
f_{25}	919.7914	3.4422	924.4339	3.8788	923.2925	3.4173	921.9507	2.8617	923.1233	3.7821	*ABGWO*
f_{26}	1091.3574	48.1577	1100.244	61.2682	1103.4981	49.925	1074.283	57.754	1101.608	56.6991	*ABGWO_V3a*
f_{27}	982.506	23.2456	982.2296	21.1732	987.1454	20.7869	982.1109	25.4186	981.046	22.1107	*ABGWO_V4a*
f_{28}	1043.4339	31.8607	1044.624	22.1775	1042.2071	32.6068	1031.1916	38.0077	1036.8447	29.1398	*ABGWO_V3a*
f_{29}	923.367	3.1454	923.6601	3.1763	922.7141	3.0531	923.1402	3.5534	922.7746	5.7176	*ABGWO_V2a*

5.2.1. KNN and K-fold cross validation

K Nearest Neighbour (KNN) is the most commonly used classification algorithm in data mining and each sample is represented by its nearest K neighbours. KNN is classified by the distance between the features of test data and training data. The computational method of distance generally selects Euclidean distance or Manhattan distance, as shown follows.

$$d(x, x') = \sqrt{\sum_{i=1}^{n}(x(i) - x'(i))^2} \tag{19}$$

$$d(x, x') = \sum_{i=1}^{n} |(x(i) - x'(i))| \tag{20}$$

where x is the training data, x' represents the test data, and n means the number of features.

K-fold cross validation randomly divides the original dataset into K parts. Among the K parts, one part is used as the test data and the other K-1 parts are the training data. It repeats the K times experiment and finally it takes the average value of the acquired K experimental results.

5.2.2. Fitness functions

In classification prediction, the classification error is usually selected as the fitness function, but for feature selection, the classification error and the number of the feature subset should be considered comprehensively. If the two algorithms acquire the same classification, the algorithm with fewer features will be selected. In the simulation, the following two fitness functions are used as evaluation criteria.

$$fitness = kfoldLoss \tag{21}$$

$$fitness = m * kfoldLoss + n * |S|/|C| \tag{22}$$

where *kfoldfLoss* is the classification error of cross validation. $|S|$ is the number of features for the subset and $|C|$ is the number of features for dataset. m and n are two coefficients, where m is 0.99 and n is 0.01 in [62].

5.2.3. Simulation analysis

In the simulation, BGWO is compared with *ABGWO*, *ABGWO_V1*, *ABGWO_V2a*, *ABGWO_V3a* and *ABGWO_V4a*. They run 20 times and 70 iterations on each dataset. Each population has 8 agents. *euclidean* and 5 are the values of parameters of *Distance* and *NumNeighbors* in KNN, and the value of *KFold* parameter in cross validation is set to 2.

Each dataset scores six compared algorithms and from the best to the worst, 1, 2, 3, 4, 5 and 6 are obtained successively. Table 17 is the statistical results acquired by using Eq. (21) as the fitness function. *Total* is the sum score obtained by each method and it can be seen that the lowest score (the optimal result) is using the ABGWO algorithm, and it gets the best results in the datasets of *Chess (King − Rook vs. King − Pawn)*, *Connectionist Bench (Sonar, Mines vs. Rocks)* and *Zoo*. BGWO's performance is the second best and its strength is in the datasets of *Breast Cancer Wisconsin (Original)*, *Tic − Tac − Toe Endgame* and *Waveform Database Generator (Version 2)*.

Table 12
The t-test results of the compared algorithms on BGWO.

Function	ABGWO	ABGWO_V1	ABGWO_V2a	ABGWO_V3a	ABGWO_V4a
f_1	+	+	+	+	+
f_2	+	+	+	+	+
f_3	+	+	+	+	+
f_4	=	=	=	=	=
f_5	=	−	−	−	−
f_6	+	+	+	+	+
f_7	+	+	+	+	+
f_8	=	−	−	−	−
f_9	+	+	+	+	+
f_{10}	+	+	+	+	+
f_{11}	+	+	+	+	+
f_{12}	+	+	+	+	+
f_{13}	=	−	−	−	−
f_{14}	=	=	=	=	=
f_{15}	=	=	=	=	=
f_{16}	=	=	=	=	=
f_{17}	=	=	=	=	=
f_{18}	=	=	=	=	=
f_{19}	=	=	=	=	=
f_{20}	=	=	=	=	=
f_{21}	=	=	=	=	=
f_{22}	=	=	=	=	=
f_{23}	=	=	=	=	−
f_{24}	+	=	=	=	=
f_{25}	+	=	=	+	=
f_{26}	+	+	+	+	+
f_{27}	+	+	+	+	+
f_{28}	+	+	+	+	+
f_{29}	+	+	+	+	+

Table 14
The convergence speed of the compared algorithms.

Function	ABGWO	ABGWO_V1	ABGWO_V2a	ABGWO_V3a	ABGWO_V4a
f_1	0.125	0.125	0.125	0.125	0.125
f_2	0.1538	0.1538	0.1538	0.1538	0.1538
f_3	0.25	0.25	0.25	0.25	0.25
f_4	1	1	1	1	1
f_5	3.1026	500	500	500	500
f_6	0.1538	0.1538	0.1538	0.1538	0.1538
f_7	0.0041	0.0041	0.0041	0.0041	0.0041
f_8	3.25	1562.5	1562.5	1562.5	1562.5
f_9	0.1304	0.087	0.087	0.087	0.087
f_{10}	0.1	0.1	0.1	0.1	0.1
f_{11}	0.2222	0.2222	0.1111	0.1111	0.1111
f_{12}	0.2	0.2	0.2	0.2	0.2
f_{13}	2.561	500	500	500	500
f_{14}	1	1	1	1	1
f_{15}	1	1	1	1	1
f_{16}	1	1	1	1	1
f_{17}	1	1	1	1	1
f_{18}	1	1	1	1	1
f_{19}	1	1	1	1	500
f_{20}	0.5	0.5	500	0.25	500
f_{21}	1	1	1	500	1
f_{22}	21	500	1	1	500
f_{23}	1	1	500	500	500
f_{24}	0.3462	500	500	500	500
f_{25}	1	0.0462	0.0462	0.0308	0.0462
f_{26}	1	0.0057	0.0057	0.0057	0.0057
f_{27}	1	0.0142	0.0142	0.0071	0.0142
f_{28}	1	0.0444	0.0444	0.0444	0.0444
f_{29}	1	0.0189	0.0189	0.0189	0.0126

ABGWO_V4a performs well in the datasets of *Statlog* (*Heart*), *Ionosphere* and *Lymphography* and ABGWO_V1 does good in the datasets of *Breast Cancer* and *SPECT Heart*. ABGWO_V3a also performs excellent in the *Congressional Voting Records*. Although ABGWO_V1, ABGWO_V2a ABGWO_V3a and ABGWO_V4a do not execute as well as BGWO. From Table 17, the number of features acquired in each dataset is less than BGWO, which is in line with the requirement of feature selection.

Table 18 shows the experimental results through using Eq. (22) as the fitness function, converted into Table 19 for the convenience of statistics. *Fitness* is calculated by using Eq. (22) according to the *Error* and *Number* of Table 18 and *Rank* is the score of the compared methods.

It can be seen that ABGWO has the best performance and ABGWO_V2a has the worst. ABGWO performs best in the datasets of *Breast Cancer Wisconsin* (*Original*), *Congressional Voting*

Table 13
The time consumption of the compared algorithms.

Function	BGWO	ABGWO	ABGWO_V1	ABGWO_V2a	ABGWO_V3a	ABGWO_V4a
f_1	9.4919	9.4978	17.5723	9.8501	10.0897	9.6345
f_2	9.4365	9.5324	17.5123	9.8138	10.0038	9.6442
f_3	10.1847	10.2227	18.2535	10.6545	10.4601	10.343
f_4	9.5886	9.5694	17.382	9.9188	9.8169	9.648
f_5	9.5829	9.58	17.3912	9.9444	9.8316	9.7118
f_6	9.5562	9.5537	17.311	9.853	9.7554	9.6365
f_7	9.6265	9.6113	17.3287	10.0373	9.772	9.6616
f_8	9.6619	9.5718	17.3831	10.0923	9.7944	9.6822
f_9	9.5534	9.5703	17.3252	10.0108	9.781	9.6447
f_{10}	9.5707	9.5268	17.3435	10.0399	9.7774	9.6616
f_{11}	9.5971	9.5495	17.3666	10.0177	9.8105	9.6949
f_{12}	9.8959	9.8638	17.6805	10.3786	10.1029	9.986
f_{13}	9.9043	9.8426	17.6836	10.4033	10.1185	9.9943
f_{14}	1.9818	1.9661	2.496	2.0022	1.9352	1.929
f_{15}	1.4193	1.4596	2.4641	1.4976	1.4494	1.4352
f_{16}	0.7251	0.7352	1.2495	0.7631	0.7421	0.7336
f_{17}	0.7233	0.7273	1.2448	0.7591	0.734	0.7252
f_{18}	0.735	0.7322	1.2501	0.7639	0.7447	0.7347
f_{19}	1.1908	1.1865	1.9634	1.2345	1.1969	1.1849
f_{20}	2.1575	2.1367	3.6803	2.1991	2.1612	2.1392
f_{21}	1.5836	1.5645	2.5914	1.6146	1.5608	1.5464
f_{22}	1.6306	1.6196	2.6522	1.6643	1.6104	1.5972
f_{23}	1.7222	1.7237	2.7401	1.7456	1.6868	1.672
f_{24}	82.9736	82.657	92.4108	83.1135	81.7925	81.6274
f_{25}	89.7022	86.6944	96.7106	87.3298	86.0383	85.957
f_{26}	85.0685	82.9012	93.0955	83.254	82.6947	82.6215
f_{27}	112.2449	109.5283	120.0003	110.1164	109.3221	109.2701
f_{28}	128.214	109.1003	120.6699	110.717	111.7604	109.1609
f_{29}	113.1182	109.2119	120.7444	110.6678	111.7754	109.3947

Table 15
The typical features to evaluate the compared algorithms.

Algorithm	Optimality	Time consumption	Convergence	Application scenarios
ABGWO	Poorness	Excellence	Poorness	Fixed-dimension function
ABGWO_V1	Excellence	Excellence	Excellence	Unimodal, Multimodal and Fixed-dimension functions
ABGWO_V2a	Generality	Poorness	Excellence	Composite multimodal functions
ABGWO_V3a	Generality	Excellence	Excellence	Unimodal, Multimodal and Fixed-dimension functions
ABGWO_V4a	Generality	Excellence	Excellence	Unimodal, Multimodal and Fixed-dimension functions

Table 16
The details of the testing datasets.

Dataset	Instances	Attributes	Attribute types	Missing values
Breast Cancer	286	9	Categorical	Yes
Breast Cancer Wisconsin (Original)	699	10	Integer	Yes
Congressional Voting Records	435	16	Categorical	Yes
Statlog (Heart)	270	13	Categorical, Real	No
Ionosphere	351	34	Integer, Real	No
Chess (King–Rook vs. King–Pawn)	3196	36	Categorical	No
Lymphography	148	18	Categorical	No
Connectionist Bench (Sonar, Mines vs. Rocks)	208	60	Real	No
SPECT Heart	267	22	Categorical	No
Tic-Tac-Toe Endgame	958	9	Categorical	No
Waveform Database Generator (Version 2)	5000	40	Real	No
Zoo	101	17	Categorical, Integer	No

Table 17
The classification errors and numbers of the compared algorithms based on Eq. (21).

Dataset	BGWO		ABGWO		ABGWO_V1		ABGWO_V2a		ABGWO_V3a		ABGWO_V4a	
	Error	Number	Error	Number	Error	Number	Error	Number	Error	Number	Error	Number
Breast Cancer	0.1912	5.25	0.1913	4.2	0.1884	4.4	0.1888	4.45	0.1886	4.3	0.1906	4.75
Breast Cancer Wisconsin (Original)	0.0214	7.3	0.0217	6.85	0.0255	5.75	0.0242	6.35	0.0229	6.25	0.0237	5.85
Congressional Voting Records	0.1568	10.35	0.1583	8.35	0.1561	8.85	0.1577	9	0.156	8.25	0.159	8.25
Statlog (Heart)	0.1835	8.4	0.1846	7.05	0.172	5.8	0.1776	6.8	0.1915	5.9	0.1702	6
Ionosphere	0.134	22.35	0.1278	16.45	0.1308	17.25	0.134	18.4	0.1285	16.15	0.1252	16.25
Chess (King–Rook vs. King–Pawn)	0.0688	27.3	0.0663	19.85	0.0892	20.35	0.088	20.9	0.0789	18	0.0723	20.5
Lymphography	0.5297	12.55	0.5324	10	0.5307	9.25	0.528	9.55	0.5301	9.15	0.5257	9.7
Connectionist Bench (Sonar, Mines vs. Rocks)	0.1885	38.95	0.1704	29.5	0.1776	30.8	0.1798	29.2	0.1873	30.45	0.1805	30.25
SPECT Heart	0.253	16.25	0.2504	12.2	0.2504	11.75	0.2571	11.65	0.2537	11.6	0.2552	11.1
Tic-Tac-Toe Endgame	0.2188	6.6	0.224	6.1	0.2315	5.9	0.2279	6	0.2289	5.8	0.2268	5.75
Waveform Database Generator (Version 2)	0.1706	32.25	0.1709	25.65	0.1757	24.15	0.1773	23.7	0.1778	22.4	0.1767	23.45
Zoo	0.0653	11.75	0.0649	9.7	0.0822	9.7	0.0683	9.15	0.0629	9.45	0.0713	10.65
Total	37		35		44		50		47		39	

Table 18
The classification errors and numbers of the compared algorithms based on Eq. (22).

Dataset	BGWO		ABGWO		ABGWO_V1		ABGWO_V2a		ABGWO_V3a		ABGWO_V4a	
	Error	Number	Error	Number	Error	Number	Error	Number	Error	Number	Error	Number
Breast Cancer	0.1952	5.3	0.1942	3.8	0.1958	4.45	0.1976	3.75	0.1941	3.95	0.2004	4.05
Breast Cancer Wisconsin (Original)	0.0353	7	0.0333	5.75	0.0345	5.45	0.0348	6	0.0348	5.5	0.033	5.35
Congressional Voting Records	0.1655	9.9	0.1598	8.25	0.1668	7.75	0.1682	8.1	0.1681	7.4	0.1643	8.5
Statlog (Heart)	0.1814	7.3	0.1738	6.5	0.1798	5.75	0.1838	6.5	0.1755	6.15	0.2003	5.9
Ionosphere	0.1405	21	0.131	16.75	0.1309	15.1	0.1309	14	0.1337	14.9	0.1363	15.75
Chess (King–Rook vs. King–Pawn)	0.084	27	0.0741	20.75	0.0807	20.6	0.1011	19.55	0.0797	19	0.0859	19.7
Lymphography	0.5376	11.85	0.5273	8.85	0.5324	8.65	0.5267	8	0.5286	7.65	0.5245	9
Connectionist Bench (Sonar, Mines vs. Rocks)	0.1978	36.6	0.1877	30.45	0.1802	28.2	0.1827	28.65	0.1843	27.25	0.1883	27.85
SPECT Heart	0.2669	15.45	0.2531	11.15	0.2614	11.65	0.2605	11.85	0.2594	10.65	0.2629	11.6
Tic-Tac-Toe Endgame	0.2254	6.8	0.2385	5.9	0.232	6	0.2434	5.3	0.2425	5.35	0.2329	5.85
Waveform Database Generator (Version 2)	0.1829	31.3	0.1762	24.35	0.1838	23.75	0.1844	24.55	0.1826	23.05	0.1821	24.1
Zoo	0.1031	11.4	0.0772	9.2	0.0761	9.45	0.0819	8.6	0.0791	9.1	0.0761	9.05

Records, *Statlog*(*Heart*), *Ionosphere*, *Chess* (*King* − *Rook* *vs*. *King* − *Pawn*), *SPECT Heart* and *Waveform Database Generator* (*Version* 2). ABGWO_V1 excels in the datasets of *Connectionist Bench* (*Sonar*, *Mines* *vs*. *Rocks*) and *Zoo*. ABGWO_V4a achieves well in the *Lymphography*. BGWO does good in the datasets of *Breast Cancer* and *Tic* − *Tac* − *Toe Endgame*.

The two evaluation criteria perform well in the datasets of *Breast cancer Wisconsin* (*original*), *Chess* (*King Room vs*. *King pawn*) and *Zoo*, and the classification errors do not exceed 10%. This is because the ratios of the first two datasets' instances/attributes are large. There are enough instances to establish the model and they have good generalization ability. Although there are not many instances of *Zoo*, its features are simple and can use few features to build classification model. The two evaluation criteria perform poorly in the *Lymphography* and the classification errors exceed 50%. Most because there is less examples and has complex structure. It performs poorly in the generalization ability of the classification model and it is difficult to establish an accurate model.

Please cite this article as: P. Hu, J. Pan and S. Chu, Improved Binary Grey Wolf Optimizer and Its application for feature selection, Knowledge-Based Systems (2020) 105746, https://doi.org/10.1016/j.knosys.2020.105746.

Table 19
The fitness of the compared algorithms.

Dataset	BGWO		ABGWO		ABGWO_V1		ABGWO_V2a		ABGWO_V3a		ABGWO_V4a	
	Fitness	Rank	Fitness	Rank	Fitness	Rank	Fitness	Rank	Fitness	Rank	Fitness	Rank
Breast Cancer	0.1912	1	0.1919	3	0.1928	4	0.1954	5	0.1917	2	0.1979	6
Breast Cancer Wisconsin (Original)	0.0286	3	0.0279	1	0.0293	5	0.0291	4	0.0296	6	0.0279	2
Congressional Voting Records	0.161	3	0.1562	1	0.1636	4	0.1648	5	0.1651	6	0.1606	2
Statlog (Heart)	0.1776	4	0.1706	1	0.1771	3	0.1806	5	0.1725	2	0.1978	6
Ionosphere	0.1356	6	0.1273	1	0.1277	2	0.1281	3	0.1306	4	0.133	5
Chess (King–Rook vs. King–Pawn)	0.0773	4	0.069	1	0.0757	3	0.0966	6	0.0752	2	0.0813	5
Lymphography	0.5364	6	0.5277	3	0.5329	5	0.5275	2	0.5297	4	0.5248	1
Connectionist Bench (Sonar, Mines vs. Rocks)	0.1936	6	0.1845	4	0.1772	1	0.1797	2	0.1816	3	0.1855	5
SPECT Heart	0.2625	6	0.2505	1	0.2587	4	0.2577	3	0.2572	2	0.2602	5
Tic-Tac-Toe Endgame	0.2201	1	0.2343	4	0.2277	2	0.24	6	0.239	5	0.2287	3
Waveform Database Generator (Version 2)	0.1768	2	0.1718	1	0.1796	5	0.1801	6	0.1786	4	0.1778	3
Zoo	0.0974	6	0.0726	3	0.0712	1	0.0776	5	0.0745	4	0.0715	2
Total		48		24		39		52		44		45

6. Conclusions

Binary Grey Wolf Optimizer solves the discretization problems of feature selection, etc. Transfer function is important as it is the key point of transforming classical GWO into binary GWO. This paper analyses the range values of *AD* of the GWO in binary condition and introduces new transfer functions. Then, a new updating equation for a parameter is proposed based on improving the abilities of exploration and exploitation. Transfer functions are improved according to the range values of *AD*. These methods show notable performance in the benchmark functions and have fast convergence, but they tend to fall into a local optimum. Towards the paper's end, KNN, cross validation and wrapper mode are used to feature selection in some datasets of UCI. The simulation results prove that the improved methods have better classification accuracy than the BGWO. The paper only uses KNN to realize feature selection. In the future, it can be combined with neural network to classify and adopts data management to decrease classification error.

CRediT authorship contribution statement

Pei Hu: Conceptualization, Methodology, Software, Writing - original draft. **Jeng-Shyang Pan:** Data curation, Methodology. **Shu-Chuan Chu:** Data curation, Methodology.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61872085), the Natural Science Foundation of Fujian Province (2018J01638) and the Fujian Provincial Department of Science and Technology (2018Y3001).

References

[1] Luanyi Yang, Zeshui Xu, Feature extraction by pca and diagnosis of breast tumors using svm with de-based parameter tuning, Int. J. Mach. Learn. Cybern. 10 (3) (2019) 591–601.
[2] Anping Zeng, Tianrui Li, Dun Liu, Junbo Zhang, Hongmei Chen, A fuzzy rough set approach for incremental feature selection on hybrid information systems, Fuzzy Sets and Systems 258 (2015) 39–60.
[3] Guoqing Li, Jing Zhao, Virginia Murray, Carol Song, Lianchong Zhang, Gap analysis on open data interconnectivity for disaster risk research, Geo-Spat. Inf. Sci. 22 (1) (2019) 45–58.
[4] Laith Mohammad Abualigah, Ahamad Tajudin Khader, Essam Said Hanandeh, A combination of objective functions and hybrid krill herd algorithm for text document clustering analysis, Eng. Appl. Artif. Intell. 73 (2018) 111–125.
[5] Jen-Da Shie, Shyi-Ming Chen, Feature subset selection based on fuzzy entropy measures for handling classification problems, Appl. Intell. 28 (1) (2008) 69–82.
[6] Li-Wei Lee, Shyi-Ming Chen, New methods for text categorization based on a new feature selection method and a new similarity measure between documents, in: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer, 2006, pp. 1280–1289.
[7] Laith Mohammad Qasim Abualigah, Feature selection and enhanced krill herd algorithm for text document clustering, in: Studies in Computational Intelligence, Springer, Boston, MA, USA, 2019, pp. 1–7.
[8] Bing Xue, Mengjie Zhang, Will N. Browne, Particle swarm optimization for feature selection in classification: A multi-objective approach, IEEE Trans. Cybern. 43 (6) (2012) 1656–1671.
[9] Yu-Tso Chen, Chi-Hua Chen, Szu Wu, Chi-Chun Lo, A two-step approach for classifying music genre on the strength of ahp weighted musical features, Mathematics 7 (1) (2019) 19.
[10] Laith Mohammad Abualigah, Ahamad Tajudin Khader, Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering, J. Supercomput. 73 (11) (2017) 4773–4795.
[11] Alper Unler, Alper Murat, A discrete particle swarm optimization method for feature selection in binary classification problems, European J. Oper. Res. 206 (3) (2010) 528–539.
[12] Harun Uğuz, A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm, Knowl.-Based Syst. 24 (7) (2011) 1024–1032.
[13] Kumar Ravi, Vadlamani Ravi, A survey on opinion mining and sentiment analysis: tasks, approaches and applications, Knowl.-Based Syst. 89 (2015) 14–46.
[14] Hui Wang, Shahryar Rahnamayan, Hui Sun, Mahamed GH Omran, Gaussian bare-bones differential evolution, IEEE Trans. Cybern. 43 (2) (2013) 634–647.
[15] Xingsi Xue, Junfeng Chen, Xin Yao, Efficient user involvement in semiautomatic ontology matching, IEEE Trans. Emerg. Topics Comput. Intell. (2018) http://dx.doi.org/10.1109/TETCI.2018.2883109.
[16] Zhenyu Meng, Jeng-Shyang Pan, Kuo-Kun Tseng, Pade: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization, Knowl.-Based Syst. 168 (2019) 80–99.
[17] Chaoli Sun, Yaochu Jin, Ran Cheng, Jinliang Ding, Jianchao Zeng, Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems, IEEE Trans. Evol. Comput. 21 (4) (2017) 644–660.
[18] Saber Salehi, Ali Selamat, M. Reza Mashinchi, Hamido Fujita, The synergistic combination of particle swarm optimization and fuzzy sets to design granular classifier, Knowl.-Based Syst. 76 (2015) 200–218.
[19] James Kennedy, Particle swarm optimization, Encyclopedia Mach. Learn. (2010) 760–766.
[20] Jin Wang, Chunwei Ju, Yu Gao, Arun Kumar Sangaiah, Gwang-jun Kim, A pso based energy efficient coverage control algorithm for wireless sensor networks, Comput. Mater. Contin. 56 (2018) 433–446.
[21] Jin Wang, Yu Gao, Wei Liu, Arun Kumar Sangaiah, Hye-Jin Kim, An improved routing schema with special clustering using pso algorithm for heterogeneous wireless sensor network, Sensors 19 (3) (2019) 671.
[22] Fahimeh Ramezani, Jie Lu, Farookh Khadeer Hussain, Task-based system load balancing in cloud computing using particle swarm optimization, Int. J. Parallel Program. 42 (5) (2014) 739–754.
[23] Fahimeh Ramezani, Jie Lu, Farookh Hussain, Task scheduling optimization in cloud computing applying multi-objective particle swarm optimization, in: International Conference on Service-Oriented Computing, Springer, 2013, pp. 237–251.
[24] Weiping Ding, Chin-Teng Lin, Zehong Cao, Deep neuro-cognitive co-evolution for fuzzy attribute reduction by quantum leaping pso with nearest-neighbor memeplexes, IEEE Trans. Cybern. 49 (7) (2018) 2744–2757.

Please cite this article as: P. Hu, J. Pan and S. Chu, Improved Binary Grey Wolf Optimizer and Its application for feature selection, Knowledge-Based Systems (2020) 105746, https://doi.org/10.1016/j.knosys.2020.105746.

[25] Shyi-Ming Chen, Chu-Han Chiou, Multiattribute decision making based on interval-valued intuitionistic fuzzy sets, pso techniques, and evidential reasoning methodology, IEEE Trans. Fuzzy Syst. 23 (6) (2014) 1905–1916.
[26] Laith Mohammad Abualigah, Ahamad Tajudin Khader, Essam Said Hanandeh, A new feature selection method to improve the document clustering using particle swarm optimization algorithm, J. Comput. Sci. 25 (2018) 456–466.
[27] Marco Dorigo, Gianni Di Caro, Ant colony optimization: a new meta-heuristic, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, Washington, USA, volume 2, 6–9 July 1999, pp. 1470–1477.
[28] Marco Dorigo, Thomas Stützle, Ant colony optimization: overview and recent advances, in: Handbook of Metaheuristics, Springer, Boston, MA, USA, ISBN: 978-3-319-91085-7, 2019, pp. 311–351.
[29] Shu-Chuan Chu, John F. Roddick, Jeng-Shyang Pan, Ant colony system with communication strategies, Inform. Sci. 167 (1–4) (2004) 63–76.
[30] Shu-Chuan Chu, Pei-Wei Tsai, Jeng-Shyang Pan, Cat swarm optimization, in: Pacific Rim International Conference on Artificial Intelligence, Guilin, China, 7–11 August 2006, pp. 854–858.
[31] Pei-Wei Tsai, Jeng-Shyang Pan, Shyi-Ming Chen, Bin-Yih Liao, Szu-Ping Hao, Parallel cat swarm optimization, in: 2008 International Conference on Machine Learning and Cybernetics, Helsinki, Finland, volume 6, 5–9 July 2008, pp. 3328–3333.
[32] Pei-Wei Tsai, Jeng-Shyang Pan, Shyi-Ming Chen, Bin-Yih Liao, Enhanced parallel cat swarm optimization based on the taguchi method, Expert Syst. Appl. 39 (7) (2012) 6309–6319.
[33] Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[34] Pei Hu, Jeng-Shyang Pan, Shu-Chuan Chu, Qing-Wei Chai, Tao Liu, Zhong-Cui Li, New hybrid algorithms for prediction of daily load of power network, Appl. Sci. 9 (21) (2019) 4514.
[35] Jeng-Shyang Pan, Pei Hu, Shu-Chuan Chu, Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power, Processes 7 (11) (2019) 845.
[36] Zhenyu Meng, Jeng-Shyang Pan, Huarong Xu, Quasi-affine transformation evolutionary (quatre) algorithm: a cooperative swarm based algorithm for global optimization, Knowl.-Based Syst. 109 (2016) 104–121.
[37] Nengxian Liu, Jeng-Shyang Pan, Jason Yang Xue, An orthogonal quasi-affine transformation evolution (o-quatre), in: Advances in Intelligent Information Hiding and Multimedia Signal Processing: Proceedings of the 15th International Conference on IIH-MSP in Conjunction with the 12th International Conference on FITAT, Springer, Boston, MA, USA, 2019, pp. 57–66.
[38] Jeng-Shyang Pan, Zhenyu Meng, Huarong Xu, Xiaoqing Li, Quasi-affine transformation evolution (quatre) algorithm: A new simple and accurate structure for global optimization, in: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer, Boston, MA, USA, 2016, pp. 657–667.
[39] Zhenyu Meng, Jeng-Shyang Pan, Quasi-affine transformation evolution with external archive (quatre-ear): an enhanced structure for differential evolution, Knowl.-Based Syst. 155 (2018) 35–53.
[40] Zheng Yu, Wenmin Wang, Learning dalts for cross-modal retrieval, CAAI Trans. Intell. Technol. 4 (1) (2019) 9–16.
[41] Kazuyuki Matsumoto, Fuji Ren, Masaya Matsuoka, Minoru Yoshida, Kenji Kita, Slang feature extraction by analysing topic change on social media, CAAI Trans. Intell. Technol. 4 (1) (2019) 64–71.
[42] Abolfazl Gharaei, Mostafa Karimi, Seyed Ashkan Hoseini Shekarabi, Joint economic lot-sizing in multi-product multi-level integrated supply chains: Generalized benders decomposition, Int. J. Syst. Sci.: Oper. Logist. (2019) 1–17.
[43] Sisi Yin, Tatsushi Nishi, Guoqing Zhang, A game theoretic model for coordination of single manufacturer and multiple suppliers with quality variations under uncertain demands, Int. J. Syst. Sci.: Oper. Logist. 3 (2) (2016) 79–91.
[44] Masoud Rabbani, Seyed Ali Akbar Hosseini-Mokhallesun, Amir Hossein Ordibazar, Hamed Farrokhi-Asl, A hybrid robust possibilistic approach for a sustainable supply chain location-allocation network design, Int. J. Syst. Sci.: Oper. Logist. (2018) 1–16.
[45] B.C. Giri, S. Bardhan, Coordinating a supply chain with backup supplier through buyback contract under supply disruption and uncertain demand, Int. J. Syst. Sci.: Oper. Logist. 1 (4) (2014) 193–204.
[46] B.C. Giri, M. Masanta, Developing a closed-loop supply chain model with price and quality dependent demand and learning in production in a stochastic environment, Int. J. Syst. Sci.: Oper. Logist. (2018) 1–17.
[47] Reza Sayyadi, Anjali Awasthi, A simulation-based optimisation approach for identifying key determinants for sustainable transportation planning, Int. J. Syst. Sci.: Oper. Logist. 5 (2) (2018) 161–174.
[48] Reza Sayyadi, Anjali Awasthi, An integrated approach based on system dynamics and anp for evaluating sustainable transportation policies, Int. J. Syst. Sci.: Oper. Logist. (2018) 1–10.
[49] Rameshwar Dubey, Angappa Gunasekaran, Sushil, Tripti Singh, Building theory of sustainable manufacturing using total interpretive structural modelling, Int. J. Syst. Sci.: Oper. Logist. 2 (4) (2015) 231–247.
[50] Nima Kazemi, Salwa Hanim Abdul-Rashid, Raja Ariffin Raja Ghazilla, Ehsan Shekarian, Simone Zanoni, Economic order quantity models for items with imperfect quality and emission considerations, Int. J. Syst. Sci.: Oper. Logist. 5 (2) (2018) 99–115.
[51] Abolfazl Gharaei, Seyed Ashkan Hoseini Shekarabi, Mostafa Karimi, Modelling and optimal lot-sizing of the replenishments in constrained, multi-product and bi-objective epq models with defective products: generalised cross decomposition, Int. J. Syst. Sci.: Oper. Logist. (2019) 1–13.
[52] Chaoqun Duan, Chao Deng, Abolfazl Gharaei, Jun Wu, Bingran Wang, Selective maintenance scheduling under stochastic maintenance quality with multiple maintenance actions, Int. J. Prod. Res. 56 (23) (2018) 7160–7178.
[53] Yuqiuge Hao, Petri Helo, Ahm Shamsuzzoha, Virtual factory system design and implementation: Integrated sustainable manufacturing, Int. J. Syst. Sci.: Oper. Logist. 5 (2) (2018) 116–132.
[54] Abolfazl Gharaei, Mostafa Karimi, Seyed Ashkan Hoseini Shekarabi, An integrated multi-product, multi-buyer supply chain under penalty, green, and quality control polices and a vendor managed inventory with consignment stock agreement: The outer approximation with equality relaxation and augmented penalty algorithm, Appl. Math. Model. 69 (2019) 223–254.
[55] Abolfazl Gharaei, Seyed Ashkan Hoseini Shekarabi, Mostafa Karimi, Ehsan Pourjavad, Alireza Amjadian, An integrated stochastic epq model under quality and green policies: generalised cross decomposition under the separability approach, Int. J. Syst. Sci.: Oper. Logist. (2019) 1–13.
[56] Masoud Rabbani, Nazanin Foroozesh, S. Meysam Mousavi, Hamed Farrokhi-Asl, Sustainable supplier selection by a new decision model based on interval-valued fuzzy sets and possibilistic statistical reference point systems under uncertainty, Int. J. Syst. Sci.: Oper. Logist. 6 (2) (2019) 162–178.
[57] Anjali Awasthi, Hichem Omrani, A goal-oriented approach based on fuzzy axiomatic design for sustainable mobility project selection, Int. J. Syst. Sci.: Oper. Logist. 6 (1) (2019) 86–98.
[58] Nita H. Shah, Urmila Chaudhari, Leopoldo Eduardo Cárdenas-Barrón, Integrating credit and replenishment policies for deteriorating items under quadratic demand in a three echelon supply chain, Int. J. Syst. Sci.: Oper. Logist. (2018) 1–12.
[59] Seyed Ashkan Hoseini Shekarabi, Abolfazl Gharaei, Mostafa Karimi, Modelling and optimal lot-sizing of integrated multi-level multi-wholesaler supply chains under the shortage and limited warehouse space: Generalised outer approximation, Int. J. Syst. Sci.: Oper. Logist. 6 (3) (2019) 237–257.
[60] S. Sarkar, B.C. Giri, Stochastic supply chain model with imperfect production and controllable defective rate, Int. J. Syst. Sci.: Oper. Logist. (2018) 1–14.
[61] Yu-Chung Tsao, Design of a carbon-efficient supply-chain network under trade credits, Int. J. Syst. Sci.: Oper. Logist. 2 (3) (2015) 177–186.
[62] Eid Emary, Hossam M. Zawbaa, Aboul Ella Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 (2016) 371–381.
[63] Qasem Al-Tashi, Said Jadid Abdul Kadir, Helmi Md Rais, Seyedali Mirjalili, Hitham Alhussian, Binary optimization using hybrid grey wolf optimization for feature selection, IEEE Access 7 (2019) 39496–39508.
[64] Lokesh Kumar Panwar, Srikanth Reddy, Ashu Verma, Bijaya K Panigrahi, Rajesh Kumar, Binary grey wolf optimizer for large scale unit commitment problem, Swarm Evol. Comput. 38 (2018) 251–266.
[65] K. Srikanth, Lokesh Kumar Panwar, Bijaya K. Panigrahi, Enrique Herrera-Viedma, Arun Kumar Sangaiah, Gai-Ge Wang, Meta-heuristic framework: quantum inspired binary grey wolf optimizer for unit commitment problem, Comput. Electr. Eng. 70 (2018) 243–260.
[66] Seyedali Mirjalili, Andrew Lewis, S-shaped versus v-shaped transfer functions for binary particle swarm optimization, Swarm Evol. Comput. 9 (2013) 1–14.
[67] Arthur Asuncion, David Newman, UCI machine learning repository, 2007.

Please cite this article as: P. Hu, J. Pan and S. Chu, Improved Binary Grey Wolf Optimizer and Its application for feature selection, Knowledge-Based Systems (2020) 105746, https://doi.org/10.1016/j.knosys.2020.105746.