



# A novel binary horse herd optimization algorithm for feature selection problem

Zahra Asghari Varzaneh<sup>1</sup> · Soodeh Hosseini<sup>1</sup> · Mohammad Masoud Javidi<sup>1</sup>

Received: 28 April 2022 / Revised: 28 July 2022 / Accepted: 27 February 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Feature selection (FS) is an essential step for machine learning problems that can improve the performance of the classification by removing useless features from the data set. FS is an NP-hard problem, so meta-heuristic algorithms can be used to find good solutions for this problem. Horse herd Optimization Algorithm (HOA) is a new meta-heuristic approach inspired by horses 'herding behavior. In this paper, an improved version of the HOA algorithm called BHOA is proposed as a wrapper-based FS method. To convert continuous to discrete search space, S-Shaped and V-Shaped transfer functions are considered. Moreover, to control selection pressure, exploration, and exploitation capabilities, the Power Distance Sums Scaling approach is used to scale the fitness values of the population. The efficiency of the proposed method is estimated on 17 standard benchmark datasets. The implementation results prove the efficiency of the proposed method based on the V-shaped category of transfer functions compared to other transfer functions and other wrapper-based FS algorithms.

**Keywords** Horse herd optimization algorithm (HOA) · Binary horse herd optimization · Transfer function · Feature selection · Classification

---

✉ Soodeh Hosseini  
so\_hosseini@uk.ac.ir

Zahra Asghari Varzaneh  
z.asghari@math.uk.ac.ir

Mohammad Masoud Javidi  
Javidi@uk.ac.ir

<sup>1</sup> Department of Computer Science, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran

# 1 Introduction

In recent years, a large number of datasets in various fields such as medicine, economics, technology, agriculture, etc., have been created. These datasets are used as the main source for knowledge discovery [30, 39]. To exploit this data, a preprocessing step must be performed on it. Then pre-processed datasets are used for data mining or other scientific intelligence algorithms. Many datasets have large dimensions that make model construction difficult and time-consuming. Also, the presence of redundant and irrelevance features in data affects the performance and accuracy of the system [12]. One of the ways to overcome these problems is using the feature selection method to reduce the size and number of features, reduce overfitting, increase classification accuracy and shorten training time. However, finding relevant/useful features is not easy due to the large search space and complex interactions between the features. Nowadays, feature selection has been considered by researchers because of its ability to improve machine learning performance and reduce problem dimensions. Different FS algorithms have been proposed to optimize the prediction model [9, 10]. Feature selection and feature reduction techniques are used in various fields including economic, social, medical, etc. In [38], the recent proliferation of big data has presented some substantial challenges and opportunities to feature selection and provided a comprehensive and structured overview of recent advances in feature selection research. Cui et al. [11], proposed a feature selection algorithm based on the correlation between Muti-Metric network traffic flow features.

In general, according to the evaluation criteria, FS methods are classified into three main groups: filter, wrapper, and embedded methods. Filter-based methods find the subset of features using statistical data dependency, information theory, mutual information, and distance. Such methods are completely independent of classification. Filtering techniques mostly consider scores for each feature and then the highest scored features are selected and assigned to a classifier for classification [29]. The feature selection process is faster in this method, but it will have poorer performance because it ignores the accuracy of the prediction in the learning algorithm. Maximum Relevance Minimum Redundancy (mRMR) [49], Multivariate Relative Discrimination Criterion (MRDC), and Relief are examples of filter-based feature selection methods [55]. Unlike filter-based methods, wrapper methods search a subset of features in the space of possible solutions using a learning algorithm as part of an assessment [34, 37]. By increasing the number of data set features, the search space is enlarged exponentially. For example, if the total number of features in a dataset is  $n$ , the number of possible solutions to obtain the feature subset is  $2^n$ . Therefore, in practice, it is impossible to achieve the optimal solution of comprehensive search techniques. Hence, feature selection based on the wrapper method is known as an NP-hard optimization problem [20]. The solution to this problem is to use meta-heuristic algorithms such as GA [33], PSO [6], ACO [35], WOA [40], GWO [15], DE [59] etc. Wrapper methods have a higher computational cost than filter-based methods but can produce better classification results, which is of major interest to researchers. Embedded-based methods, by combining the previous two methods, achieve better classification accuracy by reducing the computational cost relatively. In this method, at first, filter methods are applied to decrease the size of the data, and then, the best feature subcategory will be selected by the wrapper method [8, 28]. As a result, the desired features are less likely to be eliminated in combined methods than in filter methods.

HOA is a new meta-heuristic approach inspired by horses' herding behavior at different ages in their living environment including Grazing (G), Hierarchy (H), Sociability (S),

Imitation (I), Defense mechanism (D), and Roam (R) [43]. Despite the very good performance of the HOA algorithm for solving complex problems in high dimensions, based on the No-Free-Lunch (NFL) theorem, no optimization algorithm can solve all types of problems, and outperforms all other algorithms, while an algorithm performs well on specific datasets, it may perform poorly on similar or other datasets. This motivates us to introduce a new binary form of HOA (BHOA) for wrapper feature selection. The principal version of the HOA algorithm is provided to solve continuous problems and is unable to provide good solutions for problems of a discrete nature that have discrete search space like the FS problem. To convert continuous to discrete search space, the transfer functions are used [44]. In this paper, we propose BHOA based on four S-Shaped and four V-Shaped transfer functions for a wrapper-based FS application.

The most important reason for choosing the HOA algorithm is to provide very competitive results compared to existing powerful algorithms in solving complex problems with high dimensions up to 10,000 dimensions. In high-dimensional problems where the range of fitness values of the population is too low or too high, fitness scaling is efficient and necessary. The proposed BHOA method uses fitness scaling to convert raw fitness values that are returned by the fitness function to values in a range that are suitable for determining the velocity of horses at each iteration according to their age range. The main contributions of the paper are expressed as follows.

- Proposing an improved form of the BHOA for wrapper-based FS.
- Transferring a continuous HOA into a discrete version based on S-shaped and V-shaped transfer functions.
- Controlling and balance the selection pressure in BHOA using Power Distance Sums Scaling method.
- Testing BHOA on 17 standard benchmark datasets and comparing the efficiency of the proposed algorithm with other algorithms.

The other parts of this paper are structured as follows: Section 2 discusses a brief description of the related works. A description of the standard HOA algorithm is presented in Section 3. The details of the proposed binary approaches of HOA (BHOA) are presented in section 4, the improved BHOA algorithm based on scaling the fitness values is shown in the same section.

The implementation and results on benchmark datasets in the feature selection problem are described in Section 5. In section 6, conclusions and future works are given.

## 2 Related work

Feature selection is one of the most important concepts in machine learning. Feature selection and feature reduction techniques are used in various fields including economic, social, medical, etc. The authors in [7], proposed a hybrid principal component analysis (PCA)-firefly-based XGBoost classification model for intrusion detection in networks using GPU. Reddy et al. [50], investigated two of the prominent dimensionality reduction techniques, LDA and PCA on four popular Machine Learning using publicly available Cardiotocography datasets. Gadekallu et al. [21], used rough set theory and hybrid firefly and BAT algorithms for feature reduction and classify the disease datasets using a Fuzzy logic system. Recently, the issue of optimization has been considered by many researchers, especially in FS field. In this

section, an overview of the most recent related works on metaheuristics for the wrapper-based feature selection problem is provided.

Emary et al. [16], proposed two different procedures for creating binary gray wolf optimization for feature selection. In the first procedure, three better solutions become binary, and then crossover is performed on them. In the second approach, updated position values are used randomly to binary. In [42], the authors solved the FS problem based on the binary version of the Ant Lion Optimization (ALO) algorithm. They used six different transfer functions to map a continuous search space into a binary search space in the ALO algorithm. In 2020, an Equilibrium Optimizer (EO) based on physical concepts is introduced, which is based on physical concepts. Authors in [22] proposed two binary EO for use in the FS problem based on s-shaped and v-shaped transfer functions. The results show that BEO-V2 had better efficiency than other methods. Furthermore, binary variants of the Manta Ray Foraging Optimization (MRFO), ALO, Butterfly Optimization Algorithm (BOA), and Emperor Penguin Optimizer (EPO) are proposed in [5, 13, 17, 26], respectively. In these papers, to convert continuous into discrete search space, S-shape and V-shape transfer functions are applied.

Authors in [41], proposed a binary version of WOA for the FS problem which has used Mutation and Crossover operators for improving the exploitation phase. Also, they improved the exploration phase in the WOA algorithm with Tournament selection. The authors claim that their proposed method has superior performance. In 2019, Abdel-Basset et al. [1] introduced a new GWO algorithm to the FS problem that used Two-Step Mutation to enhance the exploitation of the GWO algorithm. In the first phase, the size of the dimensions of the dataset is reduced by maintaining the accuracy of the classification. The purpose of the second step is to add the best features that improve the accuracy of the classification. The main disadvantage of this method is that the mutation stage can be time-consuming. Ewees et al. [18] improve the performance of the MVO algorithm with chaotic maps to escape falling in the local optimal and low convergence speed. They concluded that the best chaotic map in feature selection based on MVO is a logistic map that can increase the classification accuracy and minimize the number of selected features in solving the FS problem. A new version of the FOA algorithm is introduced in [58]. The authors used the Gaussian mutation operator to escape premature convergence and the local chaotic search method to improve exploration in the algorithm. Then, they employed this algorithm to solve the feature selection problem. The results proved that the proposed method is efficient for this problem. In [25], a hybrid optimization algorithm is proposed. In this algorithm, EO is combined with Simulated Annealing (SA) to enhance the exploitation phase of EO, because SA is one of the best local search methods. Experimental results on 18 UCI datasets for solving FS problems demonstrate the efficiency of the proposed model. Recently, AL-Wajih et al. [4] proposed a method to aim to resolve the premature convergence and local stagnation problem in algorithms. They combined GWO and HHO to balance exploration and exploitation in the feature selection problem. They used a novel transfer function based on the sigmoid function to convert the continuous into binary search space.

Song et al. [52], used bare bones PSO (BBPSO) with mutual information technique for feature selection based on label correlation, making full use of the correlation between features and class labels to accelerate the convergence of swarm. Also, to enhance the exploitation performance of the algorithm, two local search operators, i.e., the supplementary operator and the deletion operator, are developed based on feature relevance-redundancy. In [32], a hybrid optimization method is proposed for feature selection, which integrated the sine-cosine algorithm (SCA) in Harris hawks optimization (HHO). The goal of SCA integration is to

cater ineffective exploration in HHO, moreover, exploitation is enhanced by dynamically adjusting candidate solutions for avoiding solution stagnancy in HHO. Punitha et al. [48], proposed a hybrid ABC approach, in which the standard ABC optimization is hybridized with a modified clonal selection algorithm of the Artificial Immune. The proposed method is used for the selection of input features and tuning of parameters of the ANN model. Authors in [51], proposed the Oppositional Crow Search Algorithm which integrates the Crow Search Algorithm (CSA) and Opposition Based Learning (OBL) method to select features and recurrent neural networks for DoS attack detection in the cloud computing environment. Table 1. Summarizes some of the research that used metaheuristic algorithms on wrapper-based feature selection problems.

By studying and reviewing the performance of the above-mentioned articles, we see that despite their good efficiency of them to get better solutions, none of them can solve all the problems related to FS. According to the NFL theorem, no optimum algorithm can solve all kinds of optimization problems and there are still many opportunities to develop or improve various optimization algorithms to find better solutions. This issue caused us to introduce a novel wrapper-based feature selection based on HOA which is discussed below.

### 3 Preliminary

#### 3.1 Horse herd optimization algorithm (HOA)

HOA is a recent meta-heuristic algorithm developed by MiarNaeimi et al. [43]. The HOA is inspired by horses' herding behavior at different ages. The horses' behavior patterns are

**Table 1** An Overview of the literature on feature selection methods

Reference	Year	Proposed Method
Emary et al. [16]	2016	A new binary version of gray wolf optimization for feature selection.
Gao et al. [22]	2020	An efficient binary equilibrium optimizer algorithm for feature selection.
Ghosh et al. [26]	2021	Using S-shaped versus V-shaped transfer functions for binary Manta ray foraging optimization in feature selection problem.
Dhiman et al. [13]	2021	A novel binary emperor penguin optimizer for automatic feature selection.
Abdel-Basset et al. [1]	2020	A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection.
Ewees et al. [18]	2019	Improving MVO with chaotic maps to escape falling in the local optimal and low convergence speed.
Zhang et al. [58]	2020	Gaussian mutational chaotic fruit fly-built optimization for feature selection.
Ghosh et al. [25]	2020	Binary Equilibrium Optimizer Combined with Simulated Annealing for Feature Selection.
AL-Wajih et al. [4]	2021	Hybrid Binary Grey Wolf With Harris Hawks Optimizer to balance exploration and exploitation in the feature selection problem.
Reddy et al. [7]	2020	A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU.
Gadekallu et al. [21]	2021	An efficient attribute reduction and fuzzy logic classifier for heart disease and diabetes prediction.
Song et al. [52]	2021	Feature selection using bare-bones particle swarm optimization with mutual information.
Punitha et al. [48]	2022	A Novel Breast Cancer Diagnosis Scheme With Intelligent Feature and Parameter Selections.

including Grazing (G), Hierarchy (H), Sociability (S), Imitation (I), Defense mechanism (D), and Roam (R).

The position of the horses in each iteration is calculated as follows:

$$X_m^{Iter, AGE} = \vec{V}_m^{Iter, AGE} + X_m^{(Iter-1), AGE}, \quad AGE = \alpha, \beta, \gamma, \delta \quad (1)$$

where,  $X_m^{Iter, AGE}$  is the position of  $m^{\text{th}}$  horse,  $Iter$  is the current iteration,  $AGE$  indicates the age range of this horse where  $\delta, \gamma, \beta, \alpha$  denote the horses at the age range of 0–5 years, 5–10 years, 10–15 years and older than 15 years, respectively.  $\vec{V}_m^{Iter, AGE}$  is the velocity vector of the considered horse. To select the age of the horses, a matrix is considered based on the best answers. As a result, 10% of the horses above the sorted matrix are selected as  $\alpha$  horses. The next 20% are in the  $\beta$  group. The next 30% are  $\gamma$  horses and the remaining 40% are  $\delta$  horses. The velocity vector of each horse according to the age range is obtained as follows:

$$\begin{aligned} \vec{V}_m^{Iter, \alpha} &= \vec{G}_m^{Iter, \alpha} + \vec{D}_m^{Iter, \alpha} \\ \vec{V}_m^{Iter, \beta} &= \vec{G}_m^{Iter, \beta} + \vec{H}_m^{Iter, \beta} + \vec{S}_m^{Iter, \beta} + \vec{D}_m^{Iter, \beta} \\ \vec{V}_m^{Iter, \gamma} &= \vec{G}_m^{Iter, \gamma} + \vec{H}_m^{Iter, \gamma} + \vec{S}_m^{Iter, \gamma} + \vec{I}_m^{Iter, \gamma} + \vec{D}_m^{Iter, \gamma} + \vec{R}_m^{Iter, \gamma} \\ \vec{V}_m^{Iter, \delta} &= \vec{G}_m^{Iter, \delta} + \vec{I}_m^{Iter, \delta} + \vec{R}_m^{Iter, \delta} \end{aligned} \quad (2)$$

The details of the HOA algorithm are discussed in the following sections.

- *Grazing (G)*

Horses are animals that graze most of their lifetime. In the HOA algorithm, the grazing area of each horse is modeled by a coefficient  $g$ , so that each horse of any age range grazes in a specific area around itself. Mathematical modeling of a horse's tendency to graze is as follows:

$$\vec{G}_m^{Iter, AGE} = g_{Iter}(\tilde{u} + \rho \tilde{l}) \left[ X_m^{(Iter-1)} \right], \quad AGE = \alpha, \beta, \gamma, \delta \quad (3)$$

$$g_m^{Iter, AGE} = g_m^{(Iter-1), AGE} \times \omega_g, \quad (4)$$

where,  $g_m^{Iter, AGE}$  decreases with the coefficient  $\omega_g$  equal to 1.5 in each iteration. The lower limit and the upper limit of grazing space is  $\tilde{l}$  and  $\tilde{u}$ , with values 0.95 and 1.05, respectively, and  $\rho$  is a random number in  $[0, 1]$ .

- *Hierarchy (H)*

Each herd of horses has a leader that the other horses follow it according to the law of hierarchy. The inclination of a herd of horses to pursue a leader is defined by the coefficient  $h$  in HOA as follows,

$$\vec{H}_m^{Iter, AGE} = h_m^{Iter, AGE} \left[ X_*^{(Iter-1)} - X_m^{(Iter-1)} \right], \quad AGE = \beta, \gamma \quad (5)$$

$$h_m^{Iter, AGE} = h_m^{(Iter-1), AGE} \times \omega_h, \quad (6)$$

where the effects of the best horse position on the velocity are defined by  $\vec{H}_m^{Iter, AGE}$ , and the position of the best horse is indicated by  $X_*^{(Iter-1)}$ .

- *Sociability (S)*

Horses usually tend to live in a social environment at the ages of 5–15 years to maintain their safety. This movement toward the mean location of other horses is indicated by parameter  $s$ , which are calculated in the following equation:

$$\vec{S}_m^{Iter, AGE} = s_m^{Iter, AGE} \left[ \left( \frac{1}{N} \sum_{j=1}^N X_j^{(Iter-1)} \right) - X_m^{(Iter-1)} \right], AGE = \beta, \gamma \quad (7)$$

$$s_m^{Iter, AGE} = s_m^{(Iter-1), AGE} \times \omega_s, \quad (8)$$

where  $N$  is the total number of horses,  $\vec{S}_m^{Iter, AGE}$  is the social motion vector of each horse and  $s_m^{Iter, AGE}$  is decreased with the coefficient  $\omega_s$  in each iteration.

- *Imitation (I)*

Young horses ( $\gamma$ ) imitate good and bad behaviors. In the HOA algorithm, this behavior is defined by factor  $i$ , which is described in Eq. 9 and 10.

$$\vec{I}_m^{Iter, AGE} = i_m^{Iter, AGE} \left[ \left( \frac{1}{pN} \sum_{j=1}^{pN} \hat{X}_j^{(Iter-1)} \right) - X^{(Iter-1)} \right], AGE = \gamma \quad (9)$$

$$i_m^{Iter, AGE} = i_m^{(Iter-1), AGE} \times \omega_i. \quad (10)$$

In these equations,  $\vec{I}_m^{Iter, AGE}$  shows the movement vector of  $i^{\text{th}}$  the horse towards the average of the best horses with  $\hat{X}$  positions.  $pN$  determines the number of horses by the best locations and  $p$  is considered as 10% of the horses.

- *Defense mechanism (D)*

Horses are exposed to predators in their habitat. The reaction of horses to the enemy is to escape from dangerous environments. The horses' defense mechanism is determined with parameter  $d$  in the HOA algorithm. This defense is shown by a negative coefficient to maintain the horse away from inappropriate locations. The escape vector of  $i^{\text{th}}$  horse from the mean of horses with worst positions are indicated by  $\vec{I}_m^{Iter, AGE}$  that described in Eq. 11 and 12.

$$\vec{D}_m^{Iter, AGE} = -d_m^{Iter, AGE} \left[ \left( \frac{1}{qN} \sum_{j=1}^{qN} \check{X}_j^{(Iter-1)} \right) - X^{(Iter-1)} \right], \quad AGE = \alpha, \beta, \gamma \quad (11)$$

$$d_m^{Iter, AGE} = d_m^{(Iter-1), AGE} \times \omega_d, \quad (12)$$

Where,  $\vec{X}$  vector shows the worst positions.  $qN$  is the number of horses with the worst positions and  $q$  is considered 20% of the horses.

- *Roam (R)*

Horses roam in nature in search of food and often go everywhere to discover new pastures. The horse is curious and suddenly goes to other places for grazing. In the HOA algorithm, this behavior is simulated randomly and determined by the parameter  $r$ . Roaming is almost seen in young horses and gradually disappears as it matures, and formulated as:

$$\vec{R}_m^{Iter, AGE} = r_m^{Iter, AGE} \rho \vec{X}^{(Iter-1)}, \quad AGE = \gamma, \delta \quad (13)$$

$$r_m^{Iter, AGE} = r_m^{(Iter-1), AGE} \times \omega_r. \quad (14)$$

$\vec{R}_m^{Iter, AGE}$  defines the random velocity vector of the  $i^{\text{th}}$  horse.

By substituting Eqs. 3 to 14 in Eq. 2, the general velocity vector equation can be obtained [43]. Figure 1 presents the flowchart of the HOA algorithm.

#### 4 The proposed binary horse herd optimization algorithm

HOA is a new algorithm inspired by horses' behavior at different ages. This algorithm has shown acceptable performance in solving complex problems with high dimensions. As

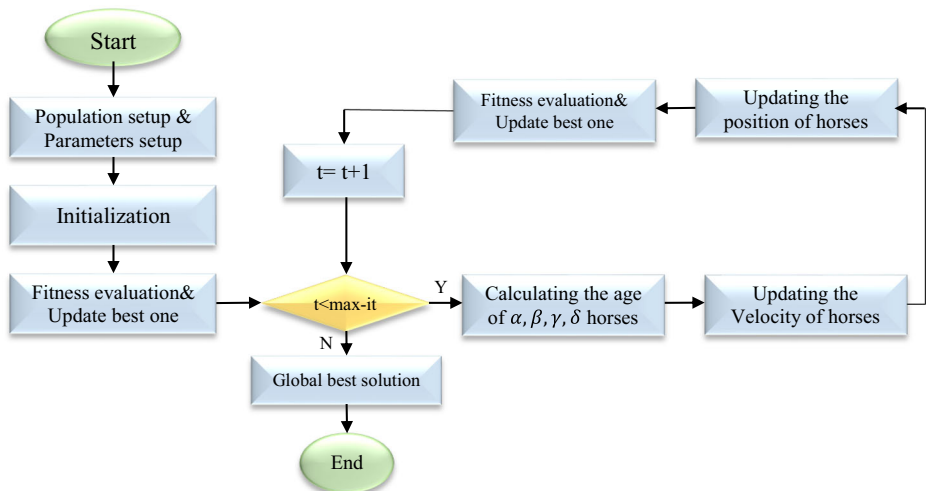


Fig. 1 Schematic of the HOA algorithm



mentioned in [43], this fast and robust algorithm has a high ability to balance the two phases of exploration and exploitation, and with high convergence rates and local optima avoidance, solves unimodal and multimodal benchmark functions with high accuracy. Also, HOA can discover the optimal solution in a very short time. The presence of these attractive features in the HOA algorithm motivates a researcher to use them in solving the wrapper-based FS problems. The standard version of the HOA algorithm is continuous, so the search space has continuous values. But FS is a binary optimization problem in which decision variables have a different format. Since in the FS problem, the variables have only values “0” and “1”, each of the solutions of the problem must be represented by the values “0” and “1”, and, a “d” dimensional search space is bounded to only these two values. To solve this problem, the binary HOA algorithm is proposed.

In the novel binary Horse herd Optimization Algorithm (BHOA) proposed in this paper, a discrete search space is considered as a hypercube in which agents are located at the corners of the hypercube, while in a continuous search space, agents move based on the position vector in the continuous real domain. Therefore, in the binary version of the HOA algorithm, position updating cannot be done according to Eq. 1. Consequently, the concept of velocity is applied to change the position of agents from “0” to “1” or conversely.

As shown in Fig. 2, to update positions, in this paper, we employ a transfer function to convert the velocity values to the probability values to specify the probability of updating the binary solution. These functions cause the HOA algorithm searches the binary space to discover the solution. The domain transfer function definition should be limited to the range [0, 1], as it indicates the probability that the position of the agent should change.

In this paper, two types of transfer functions including S-shaped and V-shaped functions are used to develop a binary form of the HOA algorithm [44]. The binary forms of the HOA algorithm are called bHOA-S and bHOA-V. In the following, each of these two proposed binary versions of HOA is discussed.

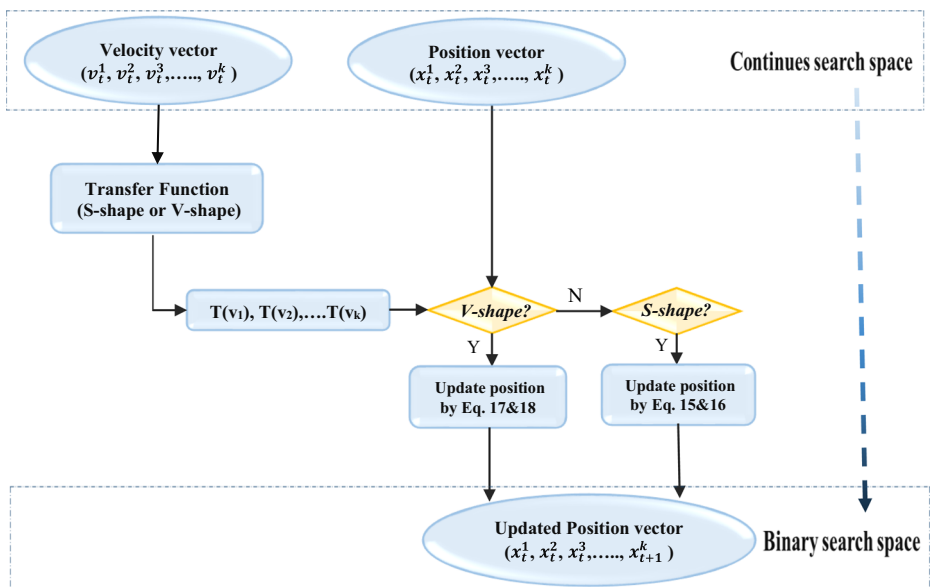


Fig. 2 Convert continuous to binary search space

#### 4.1 Binary horse herd optimization algorithm- approach1 (bHOA-S)

In this approach, the sigmoid transfer function (S-shape) is used to convert the continuous version to the binary version of the horse optimization algorithm. Therefore, updating the position of the horses according to the velocity vector and with the help of s-shaped transfer functions, compels the horses to be in a binary space and their position vector only takes values of “0” or “1”. The s-shaped function is formulated in Eq. 15 to convert the velocity vector to the probability values in the interval [0, 1].

$$S(v_i^k(t)) = \frac{1}{1 + e^{-v_i^k(t)}}, \quad (15)$$

where  $v_i^k(t)$  indicates the velocity of  $i^{\text{th}}$  horse in the  $k^{\text{th}}$  dimension at iteration  $t^{\text{th}}$ .

The calculated output is still in a continuous mode and hence it has to be thresholded to create the binary mode. The position vector of each horse, based on the probability values of their velocities obtained as follow:

$$x_i^k(t+1) = \begin{cases} 0 & \text{if } rand < S(v_i^k(t)) \\ 1 & \text{if } rand \geq S(v_i^k(t)) \end{cases}, \quad (16)$$

where,  $x_i^k(t+1)$  determines the location of the  $i^{\text{th}}$  horse at iteration  $t+1$  in the dimension  $k^{\text{th}}$ .

#### 4.2 Binary horse herd optimization algorithm- approach2 (bHOA-V)

In this approach, the V-shape transfer function is used to map the continuous to the binary form of the horse optimization algorithm. The probability values are measured using the velocity values of the horse algorithm based on V-shaped transfer functions that are calculated by:

$$V(v_i^k(t)) = |\tanh(v_i^k(t))|, \quad (17)$$

where  $v_i^k(t)$  indicates the velocity of  $i^{\text{th}}$  horse in the  $k^{\text{th}}$  dimension at iteration  $t$ .

The position vectors of the horses based on velocities are updated using the following formula:

$$x_i^k(t+1) = \begin{cases} (x_i^k(t))^{-1} & \text{if } rand < V(v_i^k(t)) \\ x_i^k(t) & \text{if } rand \geq V(v_i^k(t)) \end{cases}, \quad (18)$$

where,  $x_i^k(t+1)$  represents the position of the  $i^{\text{th}}$  horse at iteration  $t+1$  in the dimension  $k$ .

The mathematical formulas of transfer functions including (S1, S2, S3, and S4) and (V1, V2, V3, and V4) used in this paper are shown in Table 2. Furthermore, corresponding graphs of these functions are shown in Fig. 3.

#### 4.3 Improve the BHOA method based on fitness scaling

In high-dimensional problems where the range of fitness values of the population is too low or too high, fitness scaling is efficient and necessary. In these problems, if the range of defining fitness values of the population is large, the standard deviation of fitness values is high. Hence,

**Table 2** Formulation of transfer functions

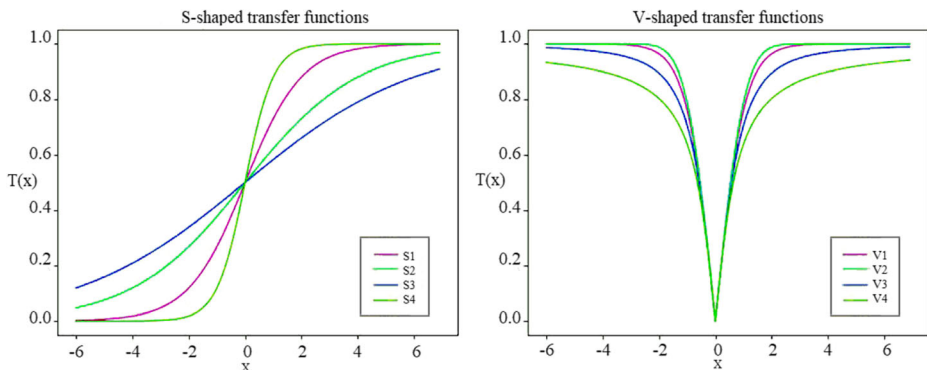
S-shaped transfer functions		V-shaped transfer functions	
S1	$T(x) = \frac{1}{1+e^x}$	V1	$T(x) =  \tanh(x) $
S2	$T(x) = \frac{1}{1+e^{2x}}$	V2	$T(x) = \left  \operatorname{erf}\left(\frac{\sqrt{\pi}}{2}x\right) \right $
S3	$T(x) = \frac{1}{1+e^{3x}}$	V3	$T(x) = \left  \frac{x}{\sqrt{1+x^2}} \right $
S4	$T(x) = \frac{1}{1+e^{\frac{x}{2}}}$	V4	$T(x) = \left  \frac{2}{\pi} \arctan\left(\frac{\pi}{2}x\right) \right $

the selection pressure is high and the probability of choosing better solutions increases and the population converges rapidly and the probability of falling in the local optimal increases. Also, if the range of the fitness values is too small, the population will not have much effect on each other, and the selection pressure is low. Consequently, the convergence rate of the algorithm is very slow. To solve this problem, the output values of the fitness function are changed by the scale functions, and then the scaled values are used in each iteration. This process will control the selection pressure, exploration, and exploitation abilities of the metaheuristic algorithms.

We use Fitness scaling to improve the efficiency of the proposed BHOA binary method in the FS problem, specifically for high-dimensional datasets. In each iteration of the BHOA, the velocity of horses in different age ranges depends on the amount of fitness values of the population. Fitness scaling changes the raw fitness values to values in the domain that is appropriate for the problem. According to the experimental results, in this paper, the best selection for scaling function is Power Distance Sums Scaling (PDSS) as follows [57]:

$$fit_i^s = \left( \sum_{fit_j \in fit_i^+} (fit_i - fit_j)^{1/\alpha} \right)^\alpha - \left( \sum_{fit_j \in fit_i^-} (fit_j - fit_i)^\alpha \right)^{1/\alpha}, \quad (19)$$

Where  $fit_i$  and  $fit_i^s$  are raw fitness values and the scaled fitness value of the  $i^{\text{th}}$  individual, respectively. Also,  $fit_i^+ = \{fit_j | fit_j > fit_i\}$  and  $fit_i^- = \{fit_j | fit_j < fit_i\}$ . The  $\alpha$  parameter is to control the standard deviation of fitness values.

**Fig. 3** The graphs of transfer functions

In some problems, the range of changes in fitness values of the individuals may be different, which may affect the efficiency of the algorithm. Therefore, in the proposed method, the scaled fitness values are used to calculate the normalized fitness value of particle  $i$  at time  $t$ , as follows:

$$Fit_i(t) = \frac{fit_i^s - fit_{worst}^s}{\sum_{j=1}^N (fit_j^s - fit_{worst}^s)} \quad (20)$$

where  $N$  determines the number of particles in each iteration and  $fit_{worst}^s$  is the lowest value of scaled fitness values.

**Theorem 1** When  $\alpha \rightarrow \infty$ , the fitness value of the best particle is “1”, and the others are “0”.

**Proof** Please see Appendix A.

**Theorem 2** When  $\alpha \rightarrow 0$ , fitness values of all the particles are equal to  $\frac{1}{N}$ .

**Proof** Please see Appendix A.

According to Theorem 1, by increasing  $\alpha$ , the standard deviation of fitness values increases. Thence, selection pressure is high. The exploitation capability of the optimization algorithm is enhanced, and the algorithm can converge to better solutions. Theorem 2 illustrates that as  $\alpha$  decreases, the standard deviation of fitness values is reduced. Therefore, the selection pressure of the algorithm is small, and the algorithm can explore the feasible area (exploration ability). In this paper, an increasing function is used to calculate the value of  $\alpha$  as follows:

$$\alpha = \eta \frac{t}{Max_t}. \quad (21)$$

In this equation,  $t$  shows time,  $Max_t$  is a maximum value of iteration and  $\eta$  is a constant value that is set to 3. Using this equation, the value of the parameter  $\alpha$  is closed to 0 at the beginning of the algorithm. The selection pressure is small and the algorithm can explore whole the search area. Towards the end of the algorithm, the value of  $\alpha$  is increased. The standard deviation of the fitness values is high. Thence, the selection pressure of the algorithm is increased and the algorithm can converge to the best solution.

#### 4.4 Improved binary HOA for feature selection

FS is considered an important process to remove useless and inefficient features and helps in enhancing classification accuracy. In this paper, an improved HOA algorithm in a binary version called (BHOA) is proposed as a wrapper method for FS problems.

Since the FS problem is a binary problem, each solution in the problem is considered as a one-dimensional vector whose vector length is equal to the dimensions of the dataset or the number of dataset features. Also, each feature in the vector has two values “0” or “1”, where the number “1” indicates the selection of a feature and the number “0” indicates the non-selection of a feature. Figure 4 shows a feature selection problem-solving vector for a dataset with  $D$  features.

For a feature vector of size  $D$ , it will be  $2^D$  different combination of features, which is a vast space of features that must be fully explored, and as the value of  $D$  increases, the number of possible solutions to the problem increases exponentially. Therefore, BHOA as a metaheuristic method is applied for the comparative search of feature space for the best feature subset. KNN (K-Nearest Neighbors) is an algorithm to classify an unknown instance into a category where most of the K-nearest neighbors are in that group [3]. In this algorithm, no model is used to find the closest neighbors and is indicated only based on the shortest distance between the test instance and the training instances. BHOA is used in a wrapper feature selection method that uses the KNN as a classifier in the evolution phase. Each feature subset is estimated to achieve the Minimum number of features with the highest classification accuracy. To create a balance between these two factors, an objective function is developed to validate the performance of the process as follows,

$$\text{objective function} = \text{Arg min}_{\vec{X}} \left( \alpha E_R + \beta \frac{|F|}{|R|} \right), \quad (22)$$

where,  $E_R$  determines the classification error rate of the KNN classifier,  $|F|$  denotes selected feature subset, and  $|R|$  is equal to the total number of features in the dataset. The vector of the dataset features is indicated by  $\vec{X} = \{x_1, x_2, \dots, x_n\}$  with values 0 and 1.  $\alpha$  and  $\beta$  are two criteria corresponding to creating a balance between the classification accuracy and feature subset size, respectively. We set  $\alpha = 0.99$  and  $\beta = 1 - \alpha$ , according to [19].

## 5 Experimental results

In this part, we perform the experiments on 17 benchmark datasets to compare the proposed BHOA approach for the FS problem with six other feature selection methods summarized as PSO [36], GA [27], GWO [45], HHO [31], DE [53] and SSA [46].

To compare the performance of the proposed BHOA algorithm with other algorithms, all of the algorithms are executed 20 times independently on a computer with Intel® 6.0 GB of RAM Core™ i5 with 2.40 GHz. Moreover, the programs are run on MATLAB 2019b platform under Windows 10.

### 5.1 Data description

To evaluate the efficiency of the proposed BHOA algorithms in comparison with other algorithms, 17 standard datasets collected from the UCI database of machine learning [14] are used in all experiments. Table 3 shows the characteristics of the used benchmark datasets including the number of features and samples of datasets. According to the size of data dimensions, the data sets are divided into three categories:

1	0	1	1	....	0	1	0
1,	2,	3,	....,				D

Fig. 4 Solution representation for the feature selection problem

**Table 3** Description of the used benchmark datasets

#	Dataset	Number of instances	Number of features
1	Arrhythmia	452	195
2	Breast cancer	638	10
3	Carcinom	174	9181
4	ClI_Sub_111	111	11,339
5	Dna	2000	180
6	Glass	214	9
7	Glioma	50	4433
8	Heart	270	13
9	Hillvalley	606	100
10	Ionosphere	351	33
11	Leukemia	72	7069
12	Lymphoma	96	4025
13	Musk	476	166
14	Sonar	208	60
15	Vehicle	700	18
16	Wine	178	13
17	Zoo	99	16

- Low-dimensional datasets
- Medium-dimensional datasets
- High dimensional datasets.

Out of 17 datasets, 4 datasets are low-dimensional (8–15 features), 5 datasets are medium-dimensional (16–100 features) and the rest 8 datasets are high-dimensional datasets (More than 100 features).

## 5.2 Parameter setting

In all conducted experiments, a wrapper approach based on the KNN classifier with  $K = 5$  is used to produce the best subset features. To learn the KNN classifier, each dataset is divided into  $K$ -fold cross-validation to validate the fitness evaluation of the algorithms.

In this approach, the dataset is randomly partitioned into  $K$  equal partition ( $k = 10$ ) where one-part (one – fold) is applied for testing, and the ( $K - 1$ ) folds are applied for the training set. All of the algorithms are run 20 independent runs and in all of them, the initial population is generated with a uniform random distribution. Furthermore, in all algorithms, the size of the population is set to 50 and the maximum number of iterations is considered 100, respectively. In feature selection problems, the lower bound is set “0” and the upper bound is set “1”. Table 4 shows the algorithms’ parameter settings the same as that of their original papers.

## 5.3 Evaluation criteria

In order to validate the efficiency of the proposed BHOA algorithm in comparison with other algorithms, a set of statistical measurements are used in this paper. These Criteria are the best, worst, and mean of fitness values; the average accuracy; the average number of selected features, standard deviation (*std*), and the average of runtime. Definition of these measurements are given as in the following:

**Table 4** Parameters setting for algorithms

Algorithm	Parameters
PSO	$C_1=2, C_2=2$ $W_1=0.4, W_2=0.9$
GA	Crossover-ratio=0.8 Mutation-ratio=0.02
GWO	$A=[2,0]$ (Linearly decreasing)
DE	CR=0.7
SSA	$C1=[2,0]$ (Linearly decreasing)
HHO	$E=[2, -2] \rightarrow 0$ (Linearly decreasing)
HOA	$w=0.99, pN=0.1, qN=0.2$

The best fitness value is the minimum fitness value obtained from 20 times the execution of each algorithm independently and  $f_*^m$  is the fitness value obtained at run number  $m$  and is given by Eq. (23).

$$\text{Best Fitness} = \min_{m=1}^{20} f_*^m. \quad (23)$$

The mean fitness value calculates the average of the fitness values obtained from 20 times the execution of each algorithm independently as follows:

$$\text{Mean fitness} = \frac{1}{20} \sum_{m=1}^{20} f_*^m. \quad (24)$$

The worst fitness value calculates the maximum of the fitness values obtained from 20 times the execution of each algorithm independently.

$$\text{Worst Fitness} = \max_{m=1}^{20} f_*^m. \quad (25)$$

The average accuracy value is the mean of the accuracy values obtained by running the algorithm 20 times as formulated by Eq. (26).  $\text{Accuracy}^m$  is the accuracy obtained from run number  $m$ .

$$\text{Average Accuracy} = \frac{1}{20} \sum_{m=1}^{20} \text{Accuracy}^m. \quad (26)$$

The average selection is the average of the number of selected features to whole features over 20 runs and is formulated by Eq. (27) where  $\text{Avg. size}^m$  is mean the number of features selected from the dataset, and  $D$  is the total number of features in the original dataset.

$$\text{Average selection} = \frac{1}{20} \sum_{m=1}^{20} \frac{\text{Avg. size}^m}{D}. \quad (27)$$

The average of the runtime of algorithms in seconds over 20 runs independently can be defined as Eq. (28) and  $\text{Runtime}^m$  is the runtime obtained at run number  $m$ .

$$\text{Average time} = \sum_{m=1}^{20} \text{Runtime}^m. \quad (28)$$

The standard deviation (std) calculates the variance of the solutions obtained from 20 times the execution of each algorithm as:

$$std = \sqrt{\frac{1}{20-1} \sum (f_*^m - mean)^2}, \quad (29)$$

where,  $f_*^m$  is the fitness value calculated at run number  $m$  and  $mean$  is the mean fitness defined in Eq. (29).

Statistical testing is required to prove the performance of the algorithms and to prove that the proposed algorithm is significantly better than other algorithms. Friedman test assigns a rank to each algorithm that the best algorithm has the first rank and the worst algorithm is assigned the last rank [24]. Moreover, the Wilcoxon rank-sum test, controls whether the experimental results of the proposed methods are statistically significantly different from other algorithms [56]. The result of this test is the p value that is used to prove the significant difference in algorithms.

## 5.4 Numerical results and discussion

This subsection shows the simulation results of the proposed binary HOA algorithm on 17 benchmark datasets. In Section 5.4.1, the efficiency of transfer functions for mapping the continuous form of the algorithm to its binary version in solving FS problems is examined and the most appropriate transfer function is discovered in terms of classification accuracy and the number of selected features. In the following, in Section 5.4.2, the results of comparing the BHOA algorithm with the other algorithms are analyzed and discussed.

### 5.4.1 Comparison of bHOA-S versus bHOA-V

In this section, we estimate the performance of the BHOA algorithm with various transfer functions including S-shaped and V-shaped functions to determine the optimal function. All these functions introduced in Table 2 are applied to the BHOA and the results of applying s1, s2, s3, and s4 (S-shaped functions) to the algorithm are called HOA-s1 to HOA-s4. Also, HOA-v1 to HOA-v4 use V-shaped functions. Moreover, in the convergence diagrams shown in Appendix B. Figure 7, BHOA1 to BHOA8 refer to the transfer functions where BHOA1 to BHOA4 are equivalent to HOA-s1 to HOA-s4 and BHOA5 to BHOA8 are equivalent to HOA-v1 to HOA-v4.

Table 5 present the mean and std. of classification accuracy of the proposed BHOA method after 20 runs on 17 datasets.

It is evident that the V-shaped functions work better than the S-shaped functions on all the datasets. The best results in the tables are indicated by bold values. In 13 out of 17 datasets, BHOA-v1 produces better results based on the mean classification accuracy than other V-shaped family algorithms. It is worth mentioning that the BHOA-v1 approach achieves 100% accuracy in 2 datasets: Leukemia and Zoo. For Lymphoma, Dna, and Sonar datasets, BHOA-v1 has achieved the second most accuracy, and for Hillvalley datasets, BHOA-v1 has achieved the third highest accuracy. Based on the comparison of std. values, BHOA-v1 compared to other algorithms has obtained favorable results. BHOA-v3 and BHOA-v4 have almost the same performance and are the second-best techniques. As we can see in Table 5, the efficiency of BHOA algorithms based on S-shaped transfer functions is almost equal in HOA-s1 to HOA-s4. The HOA-s1 algorithm performs better than the other three modes with a small difference.



**Table 5** Mean accuracy of BHOA algorithm with eight different transfer functions in 20 runs

Dataset	Measure	HOA-S1	HOA-S2	HOA-S3	HOA-S4	HOA-V1	HOA-V2	HOA-V3	HOA-V4
Arrhythmia	Mean	0.675	0.677	0.676	0.672	<b>0.692</b>	0.687	0.689	0.688
	Std	0.007	0.007	0.008	0.005	0.010	0.008	0.011	0.011
Breast cancer	Mean	0.985	0.985	0.985	0.985	0.985	0.985	0.985	0.985
	Std	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Carcinom	Mean	0.882	0.882	0.882	0.882	<b>0.917</b>	0.911	0.911	0.911
	Std	0.000	0.000	0.000	0.000	0.001	0.000	0.002	0.000
Cll_Sub_111	Mean	0.736	0.727	0.727	0.727	<b>0.793</b>	0.754	0.761	0.754
	Std	0.020	0.000	0.000	0.000	0.002	0.024	0.024	0.024
Dna	Mean	0.837	0.838	0.843	0.835	0.857	0.851	0.854	<b>0.859</b>
	Std	0.006	0.010	0.008	0.008	0.002	0.006	0.007	0.015
Glass	Mean	0.809	0.809	0.809	0.809	0.809	0.807	0.809	0.809
	Std	0.000	0.000	0.000	0.000	0.000	0.006	0.000	0.000
Glioma	Mean	0.800	0.800	0.800	0.800	<b>0.900</b>	0.860	0.860	0.900
	Std	0.000	0.000	0.000	0.000	0.009	0.005	0.005	0.000
Heart	Mean	0.833	0.833	0.831	0.833	0.833	0.833	0.833	0.830
	Std	0.000	0.000	0.005	0.000	0.000	0.000	0.000	0.007
Hillvalley	Mean	0.634	0.629	0.632	0.631	0.648	0.648	0.650	<b>0.654</b>
	Std	0.010	0.005	0.004	0.005	0.009	0.007	0.010	0.010
Ionosphere	Mean	0.931	0.931	0.931	0.931	<b>0.957</b>	0.948	0.951	0.957
	Std	0.006	0.006	0.007	0.006	0.012	0.007	0.010	0.009
Leukemia	Mean	0.985	0.957	0.987	1.000	1.000	1.000	1.000	1.000
	Std	0.031	0.032	0.000	0.000	0.000	0.000	0.000	0.000
Lymphoma	Mean	0.894	0.894	0.894	0.894	0.894	<b>0.905</b>	0.894	0.894
	Std	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.000
Musk	Mean	0.966	0.964	0.964	0.096	<b>0.978</b>	<b>0.978</b>	0.974	0.972
	Std	0.008	0.005	0.005	0.004	0.002	0.007	0.009	0.010
Sonar	Mean	0.926	0.926	0.912	0.931	0.970	0.960	<b>0.975</b>	<b>0.975</b>
	Std	0.002	0.013	0.020	0.020	0.010	0.021	0.021	0.000
Vehicle	Mean	0.784	0.784	0.777	0.781	<b>0.812</b>	0.797	0.795	0.792
	Std	0.010	0.010	0.006	0.009	0.004	0.005	0.003	0.003
Wine	Mean	0.971	0.971	0.971	0.971	0.971	0.971	0.971	0.971
	Std	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Zoo	Mean	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Std	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

As it can be observed from Table 6, BHOA-V outperformed BHOA-S on mean and std. fitness values on all the datasets. Moreover, BHOA-v1 provided better results on 11 datasets based on the mean fitness value than other V-shaped family algorithms and it achieved the second-best result on Lymphoma and Dna and it achieved the third-best result on Hillvalley, Ionosphere, and Sonar. For a better view of the efficiency of transfer functions on different datasets, the convergence curve of BHOA1 to BHOA8 algorithms is shown in Appendix B. Figure 7. In this figure, curves BHOA1 to BHOA4 show the convergence of the HOA-s1 to HOA-s4 during 100 iterations, and convergence curves HOA-v1 to HOA-v4 are shown by BHOA1 to BHOA4. According to the convergence curve of algorithms, we can see that BHOA-V algorithms are more efficient in finding the optimal solution, while BHOA-S algorithms converge to the optimal solution faster and trap into the local optimum. Therefore, they are not able to find the global optimum solution. Also, in most cases, the BHOA-v1 algorithm shows higher efficiency compared to other algorithms.

**Table 6** Mean fitness value and Std obtained from different versions of BHOA in 20 runs

Dataset	Measure	HOA-S1	HOA-S2	HOA-S3	HOA-S4	HOA-V1	HOA-V2	HOA-V3	HOA-V4
Arrhythmia	Mean	0.325	0.323	0.324	0.329	<b>0.308</b>	0.311	0.310	0.311
	Std	0.006	0.007	0.007	0.005	0.010	0.007	0.012	0.011
Breast cancer	Mean	0.019	0.019	0.019	0.019	0.019	0.019	0.019	0.019
	Std	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Carcinom	Mean	0.121	0.121	0.121	0.121	<b>0.082</b>	0.087	0.087	0.087
	Std	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000
Cll_Sub_111	Mean	0.265	0.275	0.275	0.275	0.234	0.244	<b>0.208</b>	0.244
	Std	0.021	0.000	0.000	0.000	0.002	0.023	0.023	0.023
Dna	Mean	0.165	0.164	0.159	0.167	0.143	0.151	0.148	<b>0.141</b>
	Std	0.005	0.010	0.007	0.005	0.003	0.007	0.007	0.009
Glass	Mean	0.193	0.193	0.193	0.193	0.193	0.195	0.193	0.193
	Std	0.000	0.000	0.000	0.000	0.000	0.000	0.007	0.000
Glioma	Mean	0.202	0.202	0.202	0.202	<b>0.099</b>	0.138	0.138	0.099
	Std	0.000	0.000	0.000	0.000	0.009	0.005	0.005	0.000
Heart	Mean	0.168	0.169	0.170	0.169	<b>0.168</b>	0.169	0.169	0.168
	Std	0.000	0.000	0.005	0.000	0.000	0.000	0.000	0.006
Hillvalley	Mean	0.366	0.371	0.369	0.369	0.349	0.350	0.348	<b>0.343</b>
	Std	0.009	0.005	0.004	0.005	0.010	0.007	0.013	0.010
Ionosphere	Mean	0.070	0.071	0.065	0.071	0.049	0.052	0.044	<b>0.043</b>
	Std	0.006	0.006	0.007	0.005	0.012	0.007	0.009	0.014
Leukemia	Mean	0.019	0.047	0.019	0.005	0.000	0.000	0.000	0.000
	Std	0.003	0.003	0.003	0.000	0.000	0.000	0.000	0.000
Lymphoma	Mean	0.108	0.108	0.108	0.108	0.104	<b>0.093</b>	0.104	0.104
	Std	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.002
Musk	Mean	0.038	0.040	0.040	0.040	<b>0.023</b>	0.023	0.023	0.028
	Std	0.008	0.005	0.005	0.004	0.012	0.007	0.007	0.010
Sonar	Mean	0.076	0.077	0.090	0.071	0.031	0.040	<b>0.025</b>	0.026
	Std	0.002	0.000	0.001	0.020	0.020	0.021	0.023	0.001
Vehicle	Mean	0.218	0.216	0.225	0.221	<b>0.019</b>	0.206	0.206	0.210
	Std	0.010	0.011	0.006	0.009	0.010	0.010	0.011	0.010
Wine	Mean	0.030	0.030	0.031	0.030	0.030	0.030	0.030	0.030
	Std	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Zoo	Mean	0.004	0.004	0.004	0.004	<b>0.003</b>	0.003	0.003	0.003
	Std	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

For a fair comparison of algorithms and to select the best transfer function to map the continuous form of the HOA algorithm to its discrete version, Friedman statistical test is used for ranking the algorithms. This statistical test assigns a rank to each algorithm.

Table 7 shows the result of the Friedman test according to the fitness values of the algorithms. The p value indicates whether there is a significant difference between the algorithms. We consider the significance level equal to  $\alpha = 0.05$  and if  $P$ -value  $< 0.05$ , there is a significant difference between the two algorithms. As shown in Table 7, BHOA-v1 achieves rank one and it has a better fitness value than other methods for most of the datasets. In addition, Friedman's test results show that algorithms that used V-shaped transfer functions to convert continuous to discrete space have better performance than S-shaped transfer functions. The BHOA-S3 algorithm is the weakest.

Furthermore, Table 8 shows the comparison results of the S-shaped vs. V-shaped algorithms, in which in all cases of comparison, the null hypothesis is rejected, i.e. V-shaped functions perform better than the S-shaped functions.

**Table 7** The mean rank of the algorithms based on the Friedman test

Friedman test (significance level of 0.05)			
Statistic	p value	Result	
15.80609	0.00000	H0 is rejected	
Rank		Ranking	
2.55882		Algorithm	
2.76471		HOA-V1	
2.79412		HOA-V3	
3.58824		HOA-V4	
5.52941		HOA-V2	
6.14706		HOA-S1	
6.17647		HOA-S2	
6.44118		HOA-S4	
		HOA-S3	

To determine the performance of transfer functions, the number of selected features and classification accuracy must be considered. An algorithm that achieves higher accuracy by selecting a smaller number of features is more efficient. Table 9 shows the average selection size for different versions of BHOA in 20 runs. As shown, the BHOA-V family outperformed the BHOA-S family on all the datasets. Also, HOA-v1 performed better than other algorithms on 11 datasets by providing an 84.411 average selection size.

Table 10 presents the average run time of versions of BHOA approaches which calculate the necessary time to achieve the near-optimal solutions. The results show that BHOA-V-based algorithms are better than BHOA-s-based algorithms in terms of computational time. It can be seen that BHOA-v1 has the least computational time on six datasets, whereas BHOA-v2 requires the least run time on two datasets and BHOA-v3 outperformed other algorithms on three datasets. Also, BHOA-v4 is better on the remaining six datasets.

**Table 8** Comparison of algorithms based on Friedman test

Friedman test (significance level of 0.05)			
Comparison	Statistic	Adjusted pl-value	Result
HOA-S3 vs HOA-V1	4.62092	0.00011	HO is rejected
HOA-S3 vs HOA-V3	4.37588	0.00033	HO is rejected
HOA-S3 vs HOA-V4	4.34087	0.00037	HO is rejected
HOA-S4 vs HOA-V1	4.30586	0.00042	HO is rejected
HOA-S2 vs HOA-V1	4.27085	0.00047	HO is rejected
HOA-S4 vs HOA-V3	4.06081	0.00112	HO is rejected
HOA-S4 vs HOA-V4	4.02581	0.00125	HO is rejected
HOA-S2 vs HOA-V3	4.02581	0.00125	HO is rejected
HOA-S2 vs HOA-V4	3.99080	0.00132	HO is rejected
HOA-S1 vs HOA-V1	3.53571	0.00773	HO is rejected
HOA-S3 vs HOA-V2	3.39586	0.01232	HO is rejected
HOA-S1 vs HOA-V3	3.29066	0.01699	HO is rejected
HOA-S1 vs HOA-V4	3.25565	0.01810	HO is rejected
HOA-S4 vs HOA-V2	3.08062	0.03099	HO is rejected
HOA-S2 vs HOA-V2	3.04561	0.03251	HO is rejected

**Table 9** Average the number of selected features from different versions of BHOA in 20 runs

Dataset	HOA-S1	HOA-S2	HOA-S3	HOA-S4	HOA-V1	HOA-V2	HOA-V3	HOA-V4
Arrhythmia	91.3	89.8	92	87.6	65.4	54.3	47.8	<b>52.6</b>
Breast cancer	5	5	5	5	<b>5</b>	5	5	5
Carcinom	4466.6	4534	4518.8	4520.8	408.4	276	322	<b>264.4</b>
CIJ_Sub_111	5601.2	5637.4	5628	5624.2	<b>668.4</b>	1367.8	1568.4	1815.2
Dna	91	90.8	78.6	89.4	<b>51.4</b>	66.7	83.7	52.7
Glass	4	4	4	4	<b>4</b>	4.1	4	4
Glioma	2077.6	2101.8	2079.2	2095.8	24.4	<b>22.6</b>	34.8	26
Heart	5	5.8	5	5.3	<b>4.2</b>	4.2	5	4.4
Hillvalley	49.3	44.2	49.6	47.4	<b>14.6</b>	26.2	20.6	15
Ionosphere	9.2	11.8	12	11.6	<b>4.2</b>	4.6	7	4.8
Leukemia	3439.8	3426.7	3499	3505.4	73.4	92.8	73	<b>70</b>
Lymphoma	1866.8	1891.2	1888.8	1899.6	40.6	34.2	<b>31</b>	43.2
Musk	81.6	78	78.2	80	<b>42.2</b>	43.6	73.4	58.6
Sonar	24.4	26.4	21	26	12.2	<b>9.4</b>	10.8	13
Vehicle	9.5	9.1	9.3	9	<b>8.6</b>	9.7	8.7	8.6
Wine	3.4	3.4	3.8	3.4	<b>3</b>	3.2	3	3
Zoo	7	6.8	7.4	7.4	<b>5</b>	5	5	5.8
Average	1048.982	1056.835	1057.629	1060.112	<b>84.411</b>	119.433	135.482	143.882

### 5.4.2 Comparison of BHOA with state-of-the-art approaches

Right now, the results obtained by the proposed BHOA approach are confirmed against some state-of-the-art feature selection approaches like PSO, GA, GWO, DE, SSA, and HHO. The parameter settings are the same in all the comparable algorithms and the results are obtained over 20 runs of each algorithm. To validate the efficiency of the proposed algorithm on different datasets, the datasets are categorized into three groups to verify the supremacy of the proposed algorithm on the different size datasets. Moreover, to compare the proposed algorithm with other algorithms, the binary version of HOA-v1 is used, which is called BHOA because it outperforms the other proposed types. Tables 11 and 12 outline the experimental

**Table 10** Comparison between different versions of BHOA in terms of computational time (in seconds)

Dataset	HOA-S1	HOA-S2	HOA-S3	HOA-S4	HOA-V1	HOA-V2	HOA-V3	HOA-V4
Arrhythmia	89.601	81.816	83.188	88.099	<b>72.652</b>	95.709	78.996	76.965
Breast cancer	73.074	88.058	73.514	85.431	<b>68.765</b>	69.853	71.346	68.946
Carcinom	428.388	444.965	650.617	396.442	399.133	<b>238.765</b>	326.653	287.211
CIJ_Sub_111	361.848	370.795	358.254	379.365	324.543	299.665	339.970	<b>274.563</b>
Dna	370.874	382.440	393.608	387.432	278.447	319.553	316.516	<b>261.339</b>
Glass	77.898	72.844	64.516	63.811	<b>60.768</b>	81.093	63.260	61.978
Glioma	166.480	134.910	134.031	158.664	132.664	<b>98.339</b>	123.088	118.421
Heart	64.493	63.713	93.934	64.576	<b>63.156</b>	63.436	63.808	63.476
Hillvalley	80.022	78.624	82.047	81.654	<b>72.646</b>	74.068	74.862	73.916
Ionosphere	63.494	62.663	76.517	65.650	62.189	62.867	63.886	<b>61.987</b>
Leukemia	223.483	211.434	206.027	246.418	211.203	198.432	203.533	<b>123.545</b>
Lymphoma	152.274	157.783	160.755	184.342	160.121	126.544	158.543	<b>114.330</b>
Musk	80.245	103.623	91.597	82.224	72.445	78.954	86.366	<b>72.443</b>
Sonar	62.170	59.751	60.618	64.712	58.699	59.654	<b>58.044</b>	67.659
Vehicle	67.908	65.485	67.155	67.006	<b>67.001</b>	68.443	68.654	67.231
Wine	59.661	55.142	58.264	57.596	59.654	58.071	<b>51.798</b>	55.696
Zoo	59.723	59.439	57.266	55.965	55.672	56.379	<b>54.697</b>	60.432

**Table 11** Comparison of average and std. classification accuracy BHOA with other methods in 20 runs

Dataset	Measure	PSO	GA	GWO	DE	SSA	HHO	BHOA	Full
Arrhythmia	Mean	0.702	0.713	0.717	0.710	0.644	0.697	<b>0.713</b>	0.612
	Std	0.012	0.015	0.014	0.009	0.016	0.004	0.009	
Breast cancer	Mean	0.982	0.985	0.985	0.983	0.983	0.985	<b>1.000</b>	0.643
	Std	0.000	0.000	0.003	0.003	0.003	0.000	0.000	
Carcinom	Mean	0.941	0.941	0.935	0.935	0.929	0.942	<b>0.972</b>	0.882
	Std	0.004	0.004	0.013	0.013	0.016	0.004	0.006	
Cll_Sub_111	Mean	0.736	0.735	0.727	0.727	0.718	0.718	<b>0.864</b>	0.593
	Std	0.020	0.024	0.000	0.020	0.020	0.020	0.021	
Dna	Mean	0.858	0.863	0.875	0.865	0.845	0.846	<b>0.882</b>	0.791
	Std	0.003	0.007	0.003	0.009	0.007	0.003	0.007	
Glass	Mean	0.785	0.785	0.785	0.776	0.761	0.785	<b>0.880</b>	0.642
	Std	0.000	0.000	0.000	0.021	0.000	0.000	0.000	
Glioma	Mean	0.900	0.900	0.900	0.900	0.900	0.900	<b>0.950</b>	0.805
	Std	0.000	0.000	0.000	0.000	0.000	0.000	0.003	
Heart	Mean	0.870	0.870	0.851	0.866	0.870	0.866	<b>0.926</b>	0.763
	Std	0.000	0.000	0.000	0.005	0.012	0.000	0.003	
Hillvalley	Mean	0.667	0.667	0.652	0.671	0.642	0.652	<b>0.694</b>	0.492
	Std	0.006	0.015	0.015	0.013	0.010	0.007	0.006	
Ionosphere	Mean	0.954	0.947	0.917	0.928	0.905	0.931	<b>0.968</b>	0.872
	Std	0.006	0.007	0.022	0.020	0.012	0.015	0.011	
Leukemia	Mean	0.998	1.000	0.942	0.942	0.971	0.982	<b>1.000</b>	0.853
	Std	0.000	0.000	0.031	0.031	0.039	0.031	0.000	
Lymphoma	Mean	0.894	0.894	0.894	0.894	0.894	0.894	<b>0.935</b>	0.872
	Std	0.000	0.000	0.000	0.000	0.000	0.000	0.002	
Musk	Mean	<b>0.972</b>	0.968	0.962	0.968	0.947	0.962	0.965	0.831
	Std	0.005	0.000	0.009	0.007	0.012	0.005	0.011	
Sonar	Mean	0.942	0.956	0.932	0.942	0.897	0.942	<b>0.996</b>	0.733
	Std	0.000	0.010	0.040	0.020	0.010	0.026	0.011	
Vehicle	Mean	<b>0.787</b>	0.785	0.777	0.771	0.768	0.781	0.784	0.586
	Std	0.005	0.005	0.009	0.010	0.012	0.008	0.003	
Wine	Mean	0.761	0.761	0.752	0.747	0.761	0.761	<b>0.957</b>	0.632
	Std	0.000	0.000	0.013	0.014	0.000	0.000	0.000	
Zoo	Mean	1.000	1.000	1.000	1.000	0.989	1.000	<b>1.000</b>	0.841
	Std	0.000	0.000	0.000	0.000	0.000	0.000	0.000	

results of BHOA and other competitive algorithms in terms of the mean and Std of accuracy and fitness values on 17 datasets, respectively.

As shown in Table 11, it can be found that on 15 out of 17 (88% of the datasets), BHOA has the best accuracy. On all datasets, the BHOA offers better accuracy results than others except for two datasets Musk and Vehicle. PSO outperforms other algorithms on Musk and Vehicle datasets. For these four datasets, the proposed BHOA method is the third-best algorithm with very little difference compared to competing algorithms. The last column of the table shows the accuracy without applying the FS method. Numerical results show that for high-dimensional datasets, the proposed BHOA method performs better than other algorithms with a greater difference. In addition, considering the standard deviation, the proposed BHOA algorithm, despite its high classification accuracy, is a robust method for most datasets compared to other techniques.

Table 12 shows the results achieved by seven algorithms based on the mean fitness values and Std. As it is can be observed from Table 12 that BHOA has higher performance in comparison to other methods in terms of fitness value as well as classification accuracy. The fitness values obtained from the proposed BHOA method are better than other competitive

**Table 12** Comparison of average and std. fitness value HOA with other methods in 20 runs

Dataset	Measure	PSO	GA	GWO	DE	SSA	HHO	BHOA
Arrhythmia	Mean	0.289	0.290	<b>0.282</b>	0.292	0.317	0.304	0.289
	Std	0.012	0.004	0.014	0.009	0.016	0.005	0.009
Breast cancer	Mean	0.018	0.018	0.019	0.027	0.021	0.018	<b>0.005</b>
	Std	0.000	0.000	0.002	0.003	0.002	0.000	0.000
Carcinom	Mean	0.041	0.063	0.069	0.069	0.074	0.061	<b>0.030</b>
	Std	0.000	0.000	0.001	0.001	0.006	0.000	0.003
Cll_Sub_111	Mean	0.265	0.259	0.276	0.276	0.283	0.283	<b>0.143</b>
	Std	0.021	0.024	0.000	0.031	0.020	0.019	0.021
Dna	Mean	0.145	0.139	0.131	0.131	0.155	0.157	<b>0.120</b>
	Std	0.003	0.008	0.005	0.009	0.007	0.003	0.007
Glass	Mean	0.216	0.216	0.216	0.226	0.236	0.216	<b>0.121</b>
	Std	0.000	0.000	0.000	0.021	0.021	0.000	0.000
Glioma	Mean	0.103	0.103	0.103	0.103	0.104	0.101	<b>0.071</b>
	Std	0.000	0.000	0.000	0.000	0.000	0.000	0.002
Heart	Mean	0.131	0.131	0.152	0.147	0.165	0.132	<b>0.081</b>
	Std	0.000	0.000	0.001	0.006	0.021	0.000	0.002
Hillvalley	Mean	0.323	0.333	0.350	0.331	0.357	0.358	<b>0.313</b>
	Std	0.006	0.015	0.014	0.013	0.010	0.007	0.006
Ionosphere	Mean	0.047	0.054	0.089	0.075	0.097	0.070	<b>0.029</b>
	Std	0.006	0.007	0.022	0.020	0.012	0.015	0.011
Leukemia	Mean	0.004	0.004	0.061	0.062	0.033	0.018	<b>0.000</b>
	Std	0.000	0.000	0.031	0.031	0.038	0.030	0.000
Lymphoma	Mean	0.108	0.108	0.109	0.108	0.108	0.106	<b>0.071</b>
	Std	0.000	0.000	0.000	0.000	0.000	0.000	0.002
Musk	Mean	<b>0.031</b>	0.035	0.044	0.038	0.056	0.042	0.038
	Std	0.005	0.000	0.009	0.007	0.012	0.005	0.009
Sonar	Mean	0.052	0.047	0.063	0.059	0.106	0.072	<b>0.005</b>
	Std	0.000	0.010	0.040	0.020	0.010	0.025	0.013
Vehicle	Mean	0.222	<b>0.218</b>	0.227	0.233	0.234	0.222	0.219
	Std	0.005	0.005	0.009	0.010	0.012	0.007	0.003
Wine	Mean	0.240	0.240	0.249	0.255	0.240	0.240	<b>0.040</b>
	Std	0.000	0.000	0.013	0.013	0.000	0.000	0.000
Zoo	Mean	0.003	0.003	0.005	0.005	0.014	0.004	<b>0.003</b>
	Std	0.000	0.000	0.000	0.000	0.000	0.000	0.000
<i>Rank</i>		<b>2</b>	<b>3</b>	<b>6</b>	<b>5</b>	<b>7</b>	<b>4</b>	<b>1</b>

methods on 82% of the datasets (14 out of 17). For the Arrhythmia dataset, GWO is better than other methods with a slight difference from BHOA, and BHOA achieves ranks second. On the Musk and Vehicle datasets, the PSO and GA offer better fitness values than others, respectively. Moreover, according to Appendix B. Figure 8, BHOA outperformed all state-of-the-art algorithms in convergence over all datasets except Arrhythmia, Musk, and vehicle. Table 13 outlines the average number of selected features of the proposed BHOA and other algorithms. We can see that the proposed BHOA selects fewer features than other algorithms on 15 datasets out of 17 datasets in features reduction, whereas SSA has the lowest features size over 2 datasets including Heart and Vehicle datasets. The best values are indicated by bold font. Also, the proposed algorithm achieves the first rank with an average value of 107.029 features on all datasets with a high difference from other algorithms. According to the results in Tables 11, 12, and 13, there is a significant difference in the classification accuracy and the selected features when comparing BHOA and other optimization algorithms. The selection of fewer features by the BHOA algorithm means that the proposed algorithm can explore the best

**Table 13** Comparison between BHOA and other algorithms in terms of the number of selected features

Dataset	PSO	GA	GWO	DE	SSA	HHO	BHOA
Arrhythmia	89.6	94.2	129.4	151.7	93.8	107.8	<b>62.5</b>
Breast cancer	4	6.2	6.2	5.6	5.6	4	<b>4</b>
Carcinom	4407	4463	4993	5009	4468	3073	<b>386</b>
CIJ_Sub_111	5526	5622	7151	7202	5561	5034	<b>1004</b>
Dna	87.6	83.8	135	154.4	89.4	105.4	<b>63.8</b>
Glass	4	4	4	4	4.6	4	<b>4</b>
Glioma	2022.6	2076	2158.2	2173	2113.2	1135.2	<b>45</b>
Heart	5	5	7.4	6.8	<b>5</b>	5.8	6
Hillvalley	44	42	63.2	59.8	46.8	43.2	<b>5.6</b>
Ionosphere	7.2	7.8	16	15.6	14.2	9.6	<b>4.2</b>
Leukemia	3411.6	3441.2	3680	3806.6	3464.8	2769.8	<b>33.2</b>
Lymphoma	1833.8	1880.2	1954.8	1977.6	1923.6	1087	<b>27.2</b>
Musk	72.6	75	107.6	113.2	107.6	82.6	<b>45</b>
Sonar	25	26.4	34.8	35.3	26.6	27.6	<b>14</b>
Vehicle	10.2	10.6	12.2	12.6	<b>8.8</b>	10.6	10
Wine	4	4	4.2	5	4.4	4	<b>4</b>
Zoo	6	7	8.8	9	7	7	<b>6</b>
Average	1032.953	1049.906	1203.871	1220.071	1055.553	794.741	<b>107.029</b>

areas of the search. Therefore, by selecting the best features, it improves the classification accuracy.

Table 14 reports the average computational time (in seconds) using BHOA and other algorithms. Numerical results show that the BHOA has the lowest computational time in comparison to other algorithms over 12 datasets. The SSA algorithm has the second rank with less computational time on 4 datasets, and the DE algorithm executes faster than the other algorithms on only one dataset.

The No Free Lunch theorem states that no algorithm can perform best in solving all optimization problems like FS and on all data than other algorithms. Anyway, to analyze the statistically significant difference among the experimental results, a non-parametric

**Table 14** Average computational time (in seconds) for different optimizers

Dataset	PSO	GA	GWO	DE	SSA	HHO	BHOA
Arrhythmia	54.866	83.345	56.323	54.057	54.256	93.764	<b>53.804</b>
Breast cancer	43.533	72.312	44.907	47.973	46.221	46.290	<b>42.907</b>
Carcinom	247.459	312.69	217.030	204.853	<b>170.976</b>	371.795	215.57
CIJ_Sub_111	218.995	204.242	189.7495	152.0702	<b>121.794</b>	240.041	201.135
Dna	217.361	397.021	305.683	384.335	221.140	523.642	<b>179.551</b>
Glass	41.309	62.992	47.121	39.999	<b>39.973</b>	73.631	40.346
Glioma	79.543	82.657	76.876	72.753	<b>71.390</b>	90.913	81.545
Heart	38.655	65.756	44.664	39.654	35.876	68.977	<b>34.876</b>
Hillvalley	51.856	79.765	51.967	53.543	51.398	103.445	<b>51.349</b>
Ionosphere	40.669	65.354	41.745	43.465	43.765	71.275	<b>39.884</b>
Leukemia	118.725	121.007	88.139	65.844	65.844	132.725	<b>61.543</b>
Lymphoma	100.483	102.354	85.485	69.647	71.829	140.399	<b>64.655</b>
Musk	49.984	76.876	51.476	52.142	57.607	49.993	<b>45.943</b>
Sonar	34.265	56.632	43.976	33.865	33.975	63.866	<b>32.866</b>
Vehicle	44.644	66.756	49.698	<b>40.732</b>	44.654	76.654	41.765
Wine	34.876	55.214	34.543	42.654	35.433	80.198	<b>33.220</b>
Zoo	40.002	63.345	40.737	38.153	38.543	65.765	<b>37.755</b>

Wilcoxon rank-sum test is used and  $p$  values obtained at 5% significance are recorded in Table 15. The  $p$  value  $<0.05$  proves that there is a significant difference at a level of 5%. The sign “=” in the table indicates that the results of the BHOA algorithm for the data are equal to the compared algorithm and therefore there is no significant difference in terms of performance between the compared algorithms. The sign “-” indicates a significant difference between the results, meaning that the proposed method performs better than the comparable algorithms.

Table 16a shows the statistical comparison of algorithms in general. Considering the  $p$  values calculated from the Wilcoxon statistical test and the significance level of 0.05, the proposed algorithm rejects the null hypothesis in comparison with GA, PSO, GWO, DE, SSA, and HHO algorithms, which indicates the supremacy of the proposed approach over other approaches. Also in Table 16b, the ranking results are presented based on the Friedman test, in which the proposed algorithm is in the first place.

To test the performance of the proposed algorithm BHOA, experiments are done on datasets with different dimensions. Numerical results prove the superiority of the BHOA algorithm overall data due to the classification accuracy and feature size. But in particular, the proposed algorithm works very well on high-dimensional datasets. The proposed BHOA eliminates a large number of irrelevant and useless features while increasing the accuracy of the classification and reducing runtime. Figures 5 and 6 show the high power of the BHOA algorithm on high-dimensional datasets in terms of fitness values and feature size, respectively.

To further evaluate the performance of the proposed BHOA algorithm, its simulation results are compared with other feature selection methods as well as a new metaheuristic algorithm and some of the hybrid optimization models for feature selection where the results are reported in recent related literature. Comparisons based on accuracy, fitness value, and the number of selected features are done on 6 datasets including wine, Lymphoma, Leukemia, Zoo, Ionosphere, and Sonar. The Comparative methods include MEGWO [54], GWOPSO [2], HGSO [47] and EO [23].

Table 17 shows the numerical results of the implementation of the algorithms and compares them in terms of accuracy, fitness value, and the number of selected features. According to the criterion of classification accuracy, the proposed BHOA algorithm performs better than other

**Table 15** The  $p$  value of the Wilcoxon test of BHOA and other algorithms (significant level of 0.05)

Dataset	PSO	GA	GWO	DE	SSA	HHO
Arrhythmia	4.94957E-4	0.04119	0.047545	0.03008	9.53674E-7	8.92894E-5
Breast cancer	5.36466E-4	5.36466E-4	3.8147E-9	3.8147E-9	3.8147E-9	2.267E-6
Carcinom	9.53674E-7	9.53674E-7	1.96217E-5	9.53674E-7	9.53674E-7	1.96217E-5
Cll_Sub_111	0.00539	3.28064E-4	5.2892E-5	7.62939E-6	1.90735E-6	3.23918E-5
Dna	7.62939E-6	5.2892E-5	7.62939E-6	0.00539	7.62939E-6	3.23918E-5
Glass	9.5374E-7	9.5374E-7	9.5374E-7	7.16935E-5	3.23918E-5	9.53674E-7
Glioma	3.8147E-6	3.8147E-6	3.8147E-6	3.8147E-6	3.8147E-6	3.8147E-6
Heart	1.96217E-5	1.96217E-5	3.8147E-6	1.96217E-5	9.53674E-7	1.96217E-5
Hillvalley	2.5249E-5	2.5249E-5	2.5249E-5	3.66211E-4	3.8147E-6	1.90735E-6
Ionosphere	1.90735E-6	3.8147E-6	3.99921E-5	1.90735E-6	3.99921E-5	3.99921E-5
Leukemia	5.2892E-5	=	3.23918E-5	1.90735E-6	1.90735E-6	3.23918E-5
Lymphoma	8.84622E-5	7.6239E-6	7.6239E-6	7.6239E-6	7.62939E-6	8.84622E-5
Musk	–	4.7687E-3	7.62939E-6	–	3.8147E-6	0.00571
Sonar	5.2892E-5	0.00586	3.23918E-5	0.00539	3.23918E-5	0.00571
Vehicle	–	=	3.14572E-5	7.62939E-6	7.62939E-6	1.52588E-5
Wine	7.77489E-6	7.77489E-6	9.53674E-7	1.90735E-6	7.77489E-6	7.77489E-6
Zoo	=	=	=	=	8.5535E-6	=



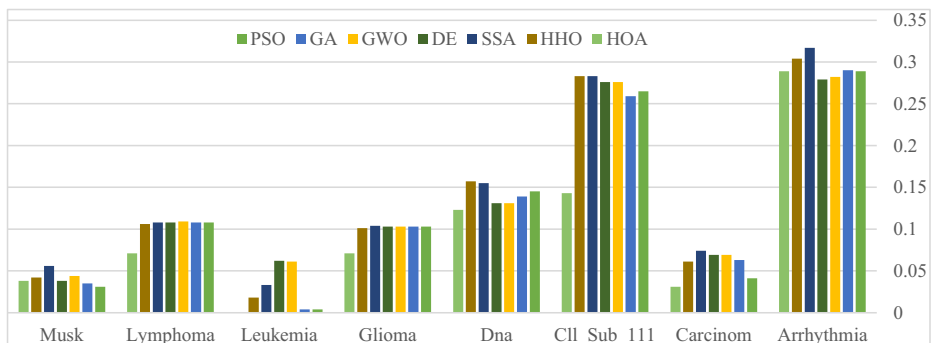
**Table 16** Statistical comparison of algorithms

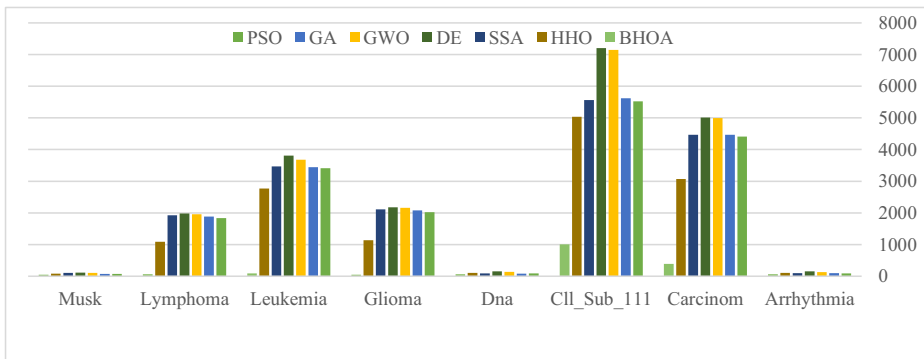
Comparison	P value	
(a) Wilcoxon test (significant level of 0.05)		
HOA vs. PSO	0.0012	
HOA vs. GA	0.0010	
HOA vs. GWO	0.0005	
HOA vs. DE	0.0007	
HOA vs. SSA	0.0002	
HOA vs. HHO	0.0002	
(b) Friedman test (Significance level of 0.05)		
Statistic	P value	Result
20.22101	0.00000	H <sub>0</sub> is rejected
Rank		Ranking Algorithm
1.41176		HOA
3.00000		PSO
3.11765		GA
4.35294		HHO
4.79412		DE
5.00000		GWO
6.32353		SSA

algorithms on all datasets except wine. Also, feature selection using the BHOA proposed algorithm caused that classification accuracy equal to 100% for zoo and Leukemia datasets. Furthermore, the value obtained for fitness shows the proposed algorithm has the highest efficiency compared to other algorithms. The number of selected features for compared algorithms shows that the proposed algorithm has selected the least number of features on 4 datasets. This algorithm has selected more features for Lymphoma and Leukemia datasets, but the classification accuracy of the BHOA algorithm on these datasets is higher than other algorithms.

Since the BHOA algorithm uses the Quick matrix sorting system, the computational complexity in the sorting section is equal to  $O(n \log n)$  and  $O(n^2)$  in the best and worst cases, respectively. Therefore, the overall computational complexity of the HOA is obtained as follows:

$$\begin{aligned}
 O(BHOA) &= O(\max_{it} \times [O(\text{Sorting}) + O(\text{position update})]) \\
 &= O(\max_{it} \times [n^2 + n \times d])
 \end{aligned} \quad (30)$$

**Fig. 5** Fitness values comparison for large datasets



**Fig. 6** The number of selected features comparison for large datasets

where  $max_{it}$  is a maximum value of iteration,  $d$  and  $n$  are also the number of problem dimensions and horses' number, respectively.

## 6 Conclusion and future works

Feature selection is considered a difficult problem in the domain of data mining to eliminate redundant and inefficient features. Metaheuristic algorithms are widely developed to solve these problems. Recently, the novel HOA algorithm has been introduced, which shows significant efficiency in solving continuous problems compared to other optimization algorithms. This algorithm can obtain solutions with high accuracy in a short time for solving continuous problems. Consequently, we are motivated to provide a binary version of this algorithm to solve discrete problems. Therefore, we propose the BHOA algorithm to solve the wrapper FS problem, which uses the KNN classifier to evaluate the solutions. We used V-shaped and S-shaped functions to change continuous space to binary search space. Moreover, to control exploration and exploitation capabilities, the Power Distance Sums Scaling approach is used to scale the fitness values of the population. The proposed method is performed on 17 data sets and 8 different transfer functions. The comparison results showed that the V-shaped functions performed better than the S-shaped functions, especially the V1 performed significantly better than other functions and has merit for use in the binary form of the HOA algorithm. Also, the proposed algorithm is compared with other optimization algorithms and the experimental results showed the supremacy of the BHOA algorithm over other algorithms in terms of the number of selected features, classification accuracy, and runtime. Despite the efficiency of meta-heuristic algorithms in solving the FS problem, the NFL theorem states that no algorithm can perform best in solving all optimization problems and there are still many opportunities to develop or improve various optimization algorithms to find better solutions.

The tendency toward the center of the search space by the probe particles can be considered as one of the limitations of HOA. In future research, we try to improve the exploration phase of this algorithm using the sensitivity analysis of motion factors. Furthermore, we can hybrid HOA with other metaheuristic algorithms. It is recommended to develop a multi-objective version of HOA for solving FS problems. Furthermore, the performance of the feature selection method can be investigated by another classifier instead of KNN such as SVM.

**Table 17** Comparison of the proposed algorithm with the state-of-the-art methods

Datasets	BHOA			MEGWO			GWOPSO			HGSO			EO		
	Acc	Fit.v	No.f	Acc	Fit.v	No.f	Acc	Fit.v	No.f	Acc	Fit.v	No.f	Acc	Fit.v	No.f
wine	0.957	0.043	4.0	0.948	0.052	4.0	1.000	0.000	6.0	0.990	0.010	5.3	0.943	0.057	6.0
Lymphoma	0.935	0.065	27.2	0.853	0.147	24.3	0.920	0.080	9.2	0.911	0.099	6.9	0.921	0.079	21.5
Leukemia	1.000	0.000	33.4	–	–	–	0.970	0.030	12.5	0.991	0.009	48.6	0.989	0.011	37.8
Zoo	1.000	0.000	6.0	0.992	0.008	5.4	1.000	0.000	6.8	1.000	0.000	5.6	1.000	0.000	6.8
Ionosphere	0.968	0.032	4.2	0.959	0.041	10.6	0.950	0.050	13.0	0.960	0.040	4.9	0.955	0.045	4.9
Sonar	0.996	0.004	14	0.948	0.052	25.6	0.960	0.040	31.2	–	–	–	0.953	0.047	16.8

Acc: Accuracy, Fit.v: Fitness value, No.f: Number of selected features

## Appendix A

Reconsider Theorem 1.

$$\lim_{\alpha \rightarrow \infty} Fit_{best} = 1.$$

**Proof** The scaled fitness values for all particles in each population can be extracted as follows.

$$\begin{aligned} \lim_{\alpha \rightarrow \infty} fit_i^s &= \left( \sum_{fit_j \in fit_i^+} (fit_i - fit_j)^{1/\alpha} \right)^\infty - \left( \sum_{fit_j \in fit_i^-} (fit_j - fit_i)^\alpha \right)^{1/\alpha} \\ &= \begin{cases} (N-1)^\infty & fit_i = fit_{best} \\ -(fit_{best} - fit_{worst}) & fit_i = fit_{worst} \\ N_i^\infty - (fit_{best} - fit_i) & otherwise \end{cases}, \end{aligned}$$

where  $N_i^-$  indicates the number of  $fit_j \in fit_i^-$ . The value of  $\lim_{\alpha \rightarrow \infty} Fit_{best}$  can be obtained as follows:

$$\begin{aligned} \lim_{\alpha \rightarrow \infty} Fit_{best} &= \lim_{\alpha \rightarrow \infty} \frac{fit_{best}^s - fit_{worst}^s}{\sum_{j=1}^N (fit_j^s - fit_{worst}^s)} \\ &= \frac{(N-1)^\infty + (fit_{best} - fit_{worst})}{(N-1)^\infty + (fit_{best} - fit_{worst}) + \sum_{i \neq best} (fit_i^s + (fit_{best} - fit_{worst}))} \\ &= \frac{1}{1 + \frac{\sum_{i \neq best} fit_i^s}{(N-1)^\infty + (fit_{best} - fit_{worst})} + \frac{(N-1)(fit_{best} - fit_{worst})}{(N-1)^\infty + (fit_{best} - fit_{worst})}} \\ &= \frac{1}{1 + \sum_{i \neq best} \frac{N_i^\infty - (fit_{best} - fit_i)}{(N-1)^\infty + (fit_{best} - fit_{worst})} + \frac{(N-1)(fit_{best} - fit_{worst})}{(N-1)^\infty + (fit_{best} - fit_{worst})}}. \end{aligned}$$

We know that  $\frac{(N-1)(fit_{best} - fit_{worst})}{(N-1)^\infty + (fit_{best} - fit_{worst})} = 0$ . Moreover  $1 > \frac{N_i^-}{N-1} \geq 0$ . Thus  $\left(\frac{N_i^-}{N-1}\right)^\infty = \frac{N_i^\infty}{(N-1)^\infty} = 0$ .

Moreover  $\frac{N_i^\infty}{(N-1)^\infty} \geq \frac{N_i^\infty - (fit_{best} - fit_i)}{(N-1)^\infty + (fit_{best} - fit_{worst})} \geq 0$ . So,  $\frac{N_i^\infty - (fit_{best} - fit_i)}{(N-1)^\infty + (fit_{best} - fit_{worst})} = 0$ .

Thus,  $\lim_{\alpha \rightarrow \infty} Fit_{best} = 1$ .

On the other hand, we know that  $\sum_{i=1}^N Fit_i(t) = 1$ . Thus, the value of other individuals is equal to "0".

Reconsider Theorem 2.

$$\lim_{\alpha \rightarrow \infty} Fit_i = \frac{1}{N}, \quad i = 1, \dots, N.$$

**Proof** The scaled fitness values for all particles in each population can be extracted as follows:

$$\begin{aligned} \lim_{\alpha \rightarrow 0^+} fit_i^s &= \left( \sum_{fit_j \in fit_i^+} (fit_i - fit_j)^0 \right)^{1/0} - \left( \sum_{fit_j \in fit_i^-} (fit_j - fit_i)^{1/0} \right)^0 \\ &= \begin{cases} (fit_{best} - fit_{worst}) & fit_i = fit_{best} \\ (N-1)^\infty & fit_i = fit_{worst} \\ (fit_i - fit_{worst}) - N_i^\infty & otherwise \end{cases}, \end{aligned}$$

where  $N_i^+$  indicates the number of  $fit_j \in fit_i^+$ . This theorem can be proved the same as theorem 1.

## Appendix B

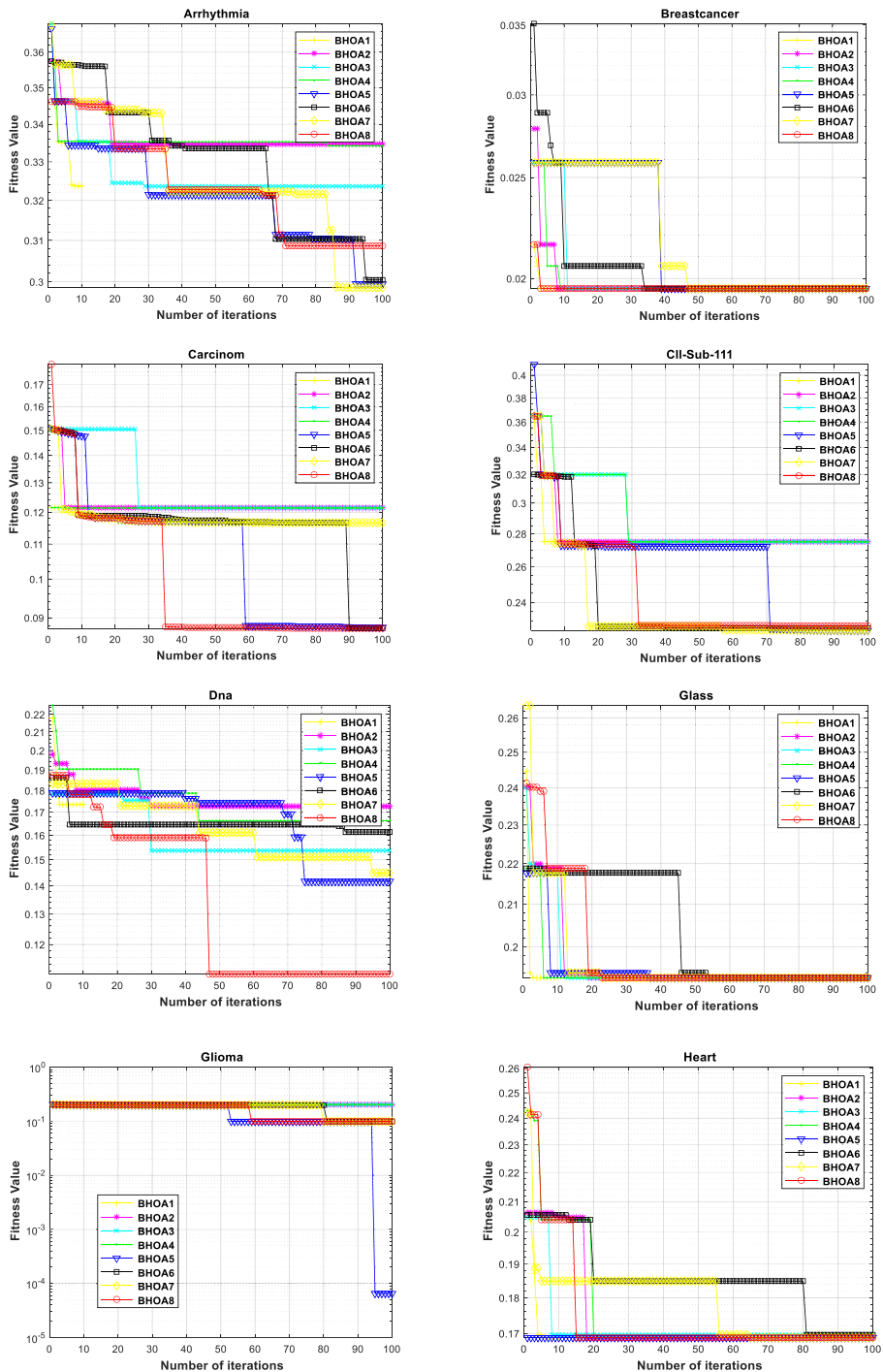


Fig. 7 Convergence comparison of different versions of BHOA

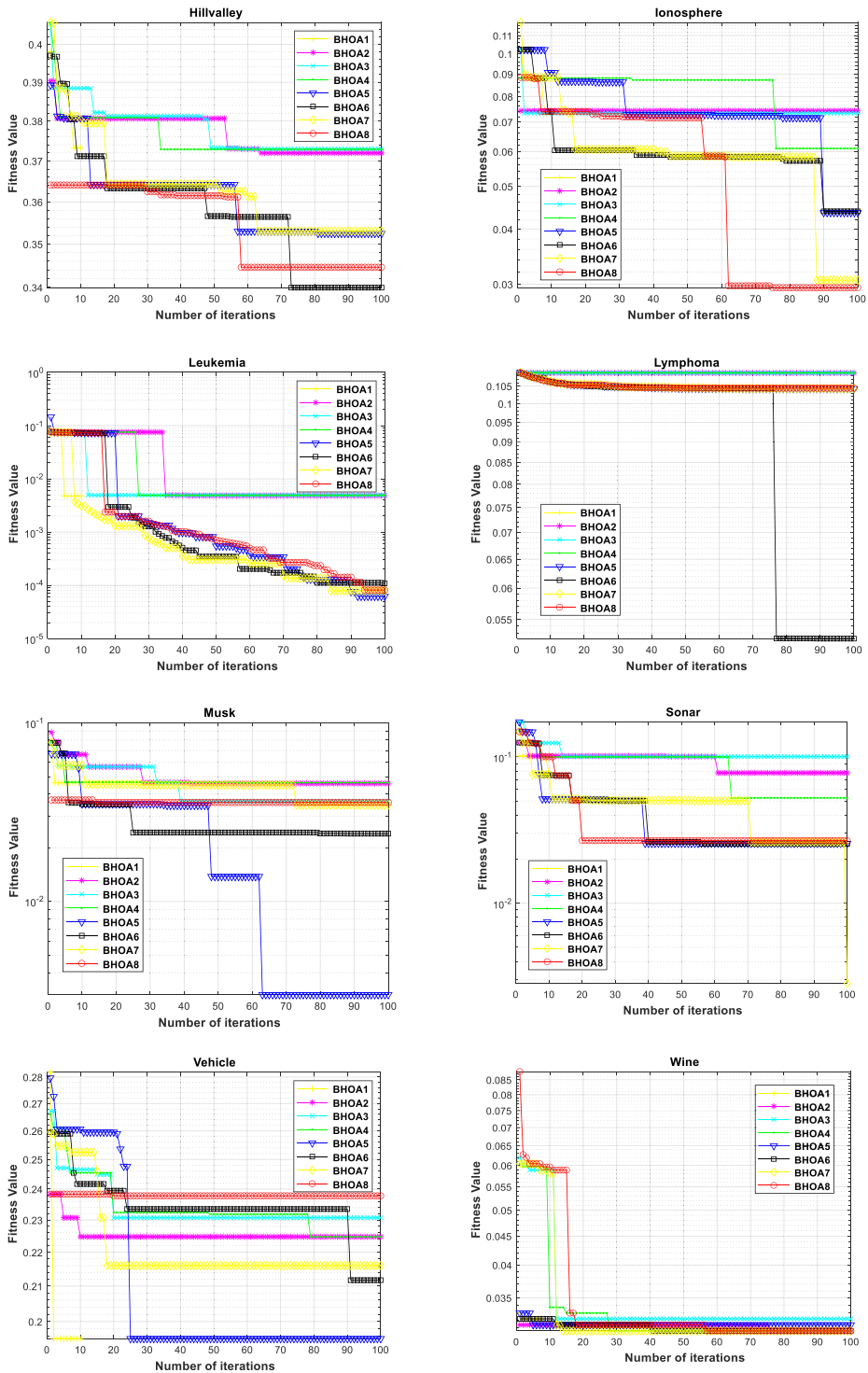
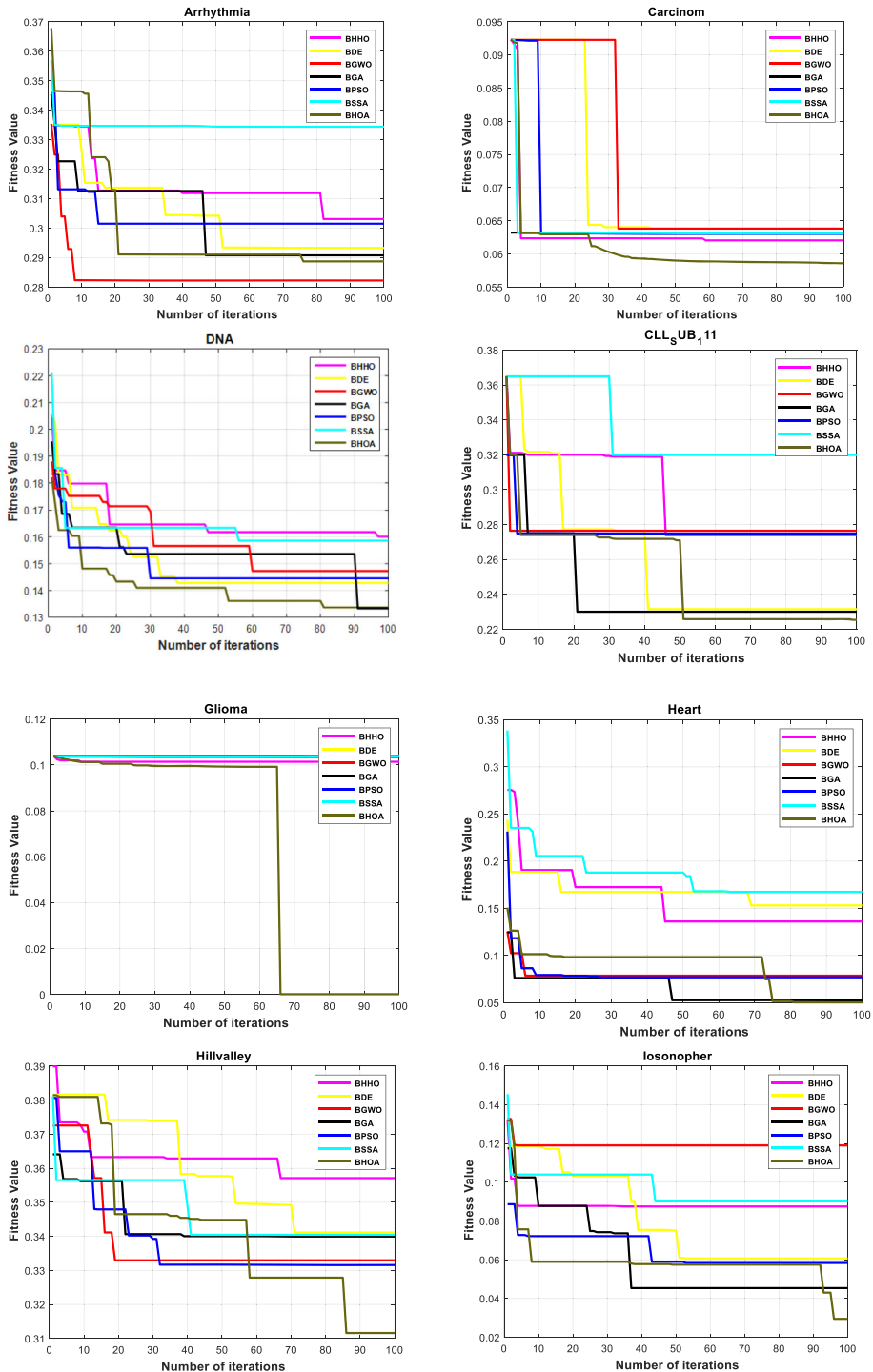


Fig. 7 (continued)



**Fig. 8** Convergence comparison of the proposed BHOA method in comparison to other algorithms

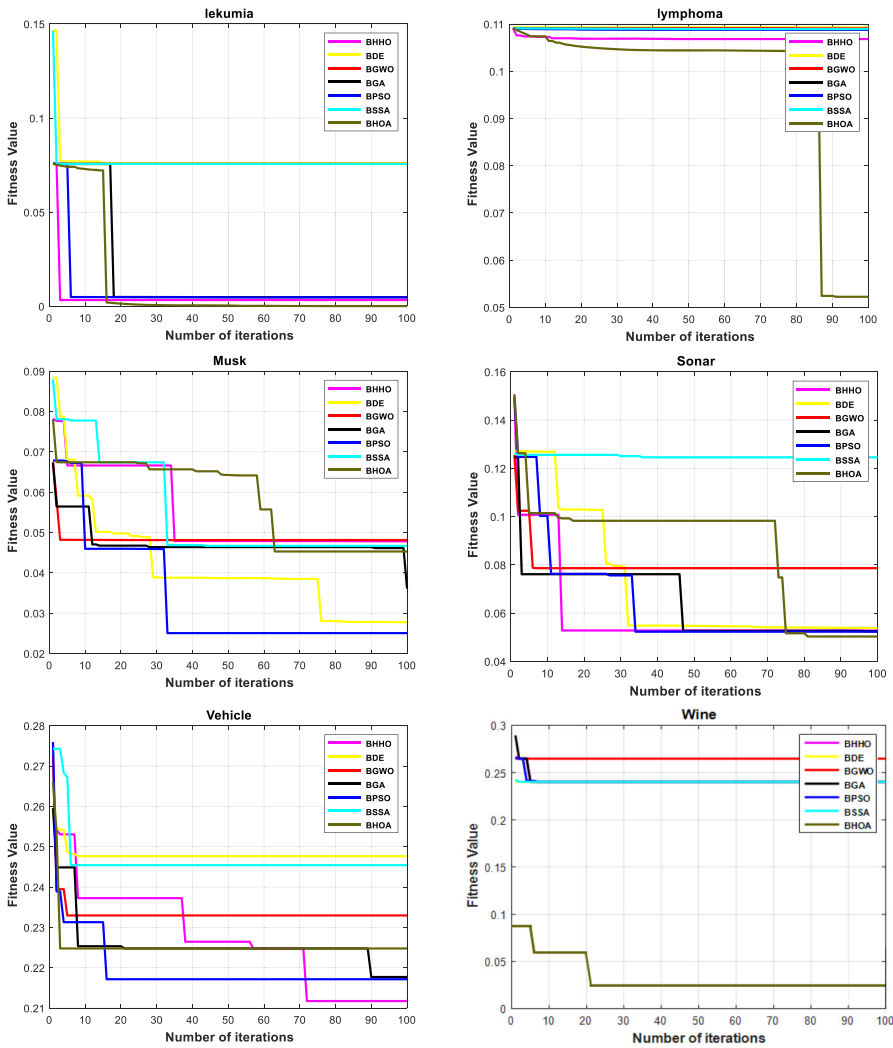


Fig. 8 (continued)

**Funding** This research received no external funding.

**Data availability** Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.



## References

1. Abdel-Basset M, El-Shahat D, El-henawy I, de Albuquerque VHC, Mirjalili SA (2020) A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Syst Appl* 139: 112824
2. Al-Tashi Q, Kadir SJA, Rais HM, Mirjalili S, Alhussian H (2019) Binary optimization using hybrid grey wolf optimization for feature selection. *IEEE Access* 7:39496–39508
3. Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. *Am Stat* 46(3): 175–185
4. Al-Wajih R, Abdulkadir SJ, Aziz N, Al-Tashi Q, Talpur N (2021) Hybrid binary Grey wolf with Harris hawks optimizer for feature selection. *IEEE Access* 9:31662–31677
5. Arora S, Anand P (2019) Binary butterfly optimization approaches for feature selection. *Expert Syst Appl* 116:147–160
6. Bello R, Gomez Y, Nowe A, Garcia MM (2007) Two-step particle swarm optimization to solve the feature selection problem. In: *Proc. 7th Int. Conf. Intell. Syst. Des. Appl. (ISDA)* pp 691–696
7. Bhattacharya S, Maddikunta PKR, Kaluri R, Singh S, Gadekallu TR, Alazab M, Tariq U (2020) A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU. *Electronics* 9(2):219
8. Blachnik M (2019) Ensembles of instance selection methods: a comparative study. *Int J Appl Math Comput Sci* 29(1):151–168
9. Chandrashekar G, Sahin F (2014) A survey on feature selection methods. *Comput Electr Eng* 40(1):16–28
10. Chizi B, Rokach L, Maimon O (2009) A Survey of Feature Selection Techniques. *Encyclopedia of Data Warehousing and Mining*, Second Edition, IGI Global. pp. 1888–1895
11. Cui Y, Dong S, Liu W (2017) Feature Selection Algorithm Based on Correlation between Muti Metric Network Traffic Flow Features. *Int Arab J Inform Technol (IAJIT)*, 14(3)
12. Dash M, Liu H (1997) Feature selection for classification. *Intel Data Analy* 1:131–156
13. Dhiman G, Oliva D, Kaur A, Singh KK, Vimal S, Sharma A, Cengiz K (2021) BEPO: a novel binary emperor penguin optimizer for automatic feature selection. *Knowl-Based Syst* 211:106560
14. Dua D, KarraTaniskidou E (2017) UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, <http://archive.ics.uci.edu/ml>
15. Emary E, Zawbaa HM, Hassanien AE (2016) Binary grey wolf optimization approaches for feature selection. *Neurocomputing* 172:371–381
16. Emary E, Zawbaa HM, Hassanien AE (2016) Binary grey wolf optimization approaches for feature selection. *Neurocomputing* 172:371–381
17. Emary E, Zawbaa HM, Hassanien A (2016) Binary ant lion approaches for feature selection. *Neurocomputing* 213:54–65
18. Ewees AA, Abd El Aziz M, Hassanien AE (2019) Chaotic multi-verse optimizer-based feature selection. *Neural Comput Applic* 31(4):991–1006
19. Faris H, Mafarja MM, Heidari AA, Aljarah I, al-Zoubi A'M, Mirjalili S, Fujita H (2018) An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowl-Based Syst* 154:43–67
20. Fayyad U, Piatetsky-Shapiro G, Smyth P (1996) From data mining to knowledge discovery in databases. *AI Mag* 17(3):37
21. Gadekallu TR, Gao X-Z (2021) An efficient attribute reduction and fuzzy logic classifier for heart disease and diabetes prediction. *Recent Adv Comput Sci Commun* 14(1):158–165
22. Gao Y, Zhou Y, Luo Q (2020) An efficient binary equilibrium optimizer algorithm for feature selection. *J IEEE Access* 8:140936–140963
23. Gao Y, Zhou Y, Luo Q (2020) An efficient binary equilibrium optimizer algorithm for feature selection. *IEEE Access* 8:140936–140963
24. García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. *Inf Sci* 180(10):2044–2064
25. Ghosh KK, Guha R, Bera SK, Sarkar R, Mirjalili S (2020) BEO: binary equilibrium optimizer combined with simulated annealing for feature selection
26. Ghosh KK, Guha R, Bera SK, Kumar N, Sarkar R, Applications (2021) S-shaped versus V-shaped transfer functions for binary Manta ray foraging optimization in feature selection problem, *Neural Computing and Applications*. pp. 1–15
27. Goldberg DE (1989) *Genetic algorithms in search, optimization, and machine learning*. Reading: Addison-Wesley
28. Guha R, Ghosh M, Mutsuddi S, Sarkar R, Mirjalili S (2020) Embedded chaotic whale survival algorithm for filter-wrapper feature selection. *Soft Comput* 24(17):12821–12843

29. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. *J Mach Learn Res* 3:1157–1182
30. Han J, Pei J, Kamber M (2011) *Data Mining: Concepts and Techniques*. Amsterdam, the Netherlands. Elsevier
31. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. *Futur Gener Comput Syst* 97:849–872
32. Hussain K, Neggaz N, Zhu W, Houssein EH (2021) An efficient hybrid sine-cosine Harris hawks optimization for low and high-dimensional feature selection. *Expert Syst Appl* 176:114778
33. Kabir MM, Shahjahan M, Murase K (2011) A new local search based hybrid genetic algorithm for feature selection. *Neurocomputing* 74(17):2914–2928
34. Karegowda AG, Manjunath AS, Jayaram MA (Feb. 2010) Feature subset selection problem using wrapper approach in supervised learning. *Int J Comput Appl* 1(7):13–17
35. Kashef S, Nezamabadi-pour H (2015) An advanced ACO algorithm for feature subset selection. *Neurocomputing* 147:271–279
36. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of ICNN'95-international conference on neural networks*. vol. 4, pp. 1942–1948: IEEE
37. Kohavi R, John GH (1997) Wrappers for feature subset selection. *Artif Intell* 97(1–2):273–324
38. Li J, Cheng K, Wang S (2017) Feature selection: A data perspective. *ACM Comput Surv (CSUR)* 50(6):1–45
39. Liu H, Motoda H (2012) *Feature Selection for Knowledge Discovery and Data Mining*. Springer 454
40. Mafarja M, Mirjalili S (2018) Whale optimization approaches for wrapper feature selection. *Appl Soft Comput* 62:441–453
41. Mafarja M, Mirjalili SA (2018) Whale optimization approach for wrapper feature selection. *Appl Soft Comput* 62:441–453
42. Mafarja M, Eleyan D, Abdullah S, Mirjalili S (2017) S-shaped vs. V-shaped transfer functions for ant lion optimization algorithm in feature selection problem. In: *Proceedings of the international conference on future networks and distributed systems*. pp. 1–7
43. MiarNaeimi F, Azizyan G, Rashki M (2021) Horse herd optimization algorithm: a nature-inspired algorithm for high-dimensional optimization problems. *Knowl-Based Syst* 213:106711
44. Mirjalili S, Lewis A (2013) S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm Evol Comput* 9:1–14
45. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
46. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw*. PP. 114, PP. 163–191
47. Neggaz N, Houssein EH, Hussain K (2020) An efficient henry gas solubility optimization for feature selection. *Expert Syst Appl* 152:113364
48. Punitha S, Stephan T, Gandomi AH, P. I. Biomedicine (2022) A Novel Breast Cancer Diagnosis Scheme With Intelligent Feature and Parameter Selections. *Computer Methods and Programs in Biomedicine* 214: 106432
49. Ramírez-Gallego S, Lastra I, Martínez-Rego D, *et al* (2016) Fast-mRMR: Fast Minimum Redundancy Maximum Relevance Algorithm for High-Dimensional Big Data. *Int J Intel Syst* vol. 32, no. 2
50. Reddy GT, Reddy MPK, Lakshmana K, Kaluri R, Rajput DS, Srivastava G, Baker T (2020) Analysis of dimensionality reduction techniques on big data. *IEEE Access* 8:54776–54788
51. SaiSindhuTheja R, Shyam GK (2021) An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment. *Appl Soft Comput* 100: 106997
52. Song X-f, Zhang Y, Gong D-w, Sun X-y (2021) Feature selection using bare-bones particle swarm optimization with mutual information. *Pattern Recogn* 112:107804
53. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11(4):341–359
54. Tu Q, Chen X, Liu X (2019) Multi-strategy ensemble grey wolf optimizer and its application to feature selection. *Appl Soft Comput* 76:16–30
55. Vergara JR, Estévez PA (Jan. 2014) A review of feature selection methods based on mutual information. *Neural Comput Applic* 24(1):175–186
56. Wilcoxon F (1945) Individual comparisons by ranking methods. *Biom Bull* 6:80–83

57. Zhang LF, Zhou CX, He R, Xu Y, Yan ML (2015) A novel fitness allocation algorithm for maintaining a constant selective pressure during GA procedure. *Neurocomputing* 148:3–16
58. Zhang X, Xu Y, Yu C, Heidari AA, Li S, Chen H, Li C (2020) Gaussian mutational chaotic fruit fly-built optimization and feature selection. *Expert Syst Appl* 141:112976
59. Zorarpacı E, Özel SA (2016) A hybrid approach of differential evolution and artificial bee colony for feature selection. *Expert Syst Appl* 62:91–103

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.