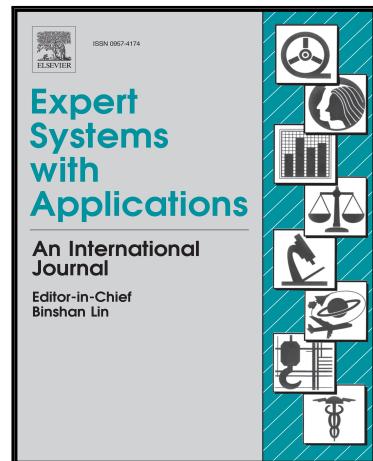


# Accepted Manuscript

Binary butterfly optimization approaches for feature selection

Sankalap Arora, Priyanka Anand

PII: S0957-4174(18)30566-9  
DOI: <https://doi.org/10.1016/j.eswa.2018.08.051>  
Reference: ESWA 12182



To appear in: *Expert Systems With Applications*

Received date: 27 March 2018  
Revised date: 12 August 2018  
Accepted date: 29 August 2018

Please cite this article as: Sankalap Arora, Priyanka Anand, Binary butterfly optimization approaches for feature selection, *Expert Systems With Applications* (2018), doi: <https://doi.org/10.1016/j.eswa.2018.08.051>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

**Highlights**

- Two binary variants of the butterfly optimization algorithm (BOA) are proposed.
- Two transfer functions are used to map the continuous search space to discrete one.
- Twenty one UCI datasets are utilized in the experiments.
- A superior performance of the proposed binary variants is proved in the experiments.

# Binary butterfly optimization approaches for feature selection

Sankalap Arora<sup>a,\*</sup>, Priyanka Anand<sup>b</sup>

<sup>a</sup>*DAV University, Jalandhar, Punjab, India*

<sup>b</sup>*Lovely Professional University, Jalandhar, Punjab, India*

## Abstract

In this paper, binary variants of the Butterfly Optimization Algorithm (BOA) are proposed and used to select the optimal feature subset for classification purposes in a wrapper-mode. BOA is a recently proposed algorithm that has not been systematically applied to feature selection problems yet. BOA can efficiently explore the feature space for optimal or near-optimal feature subset minimizing a given fitness function. The two proposed binary variants of BOA are applied to select the optimal feature combination that maximizes classification accuracy while minimizing the number of selected features. In these variants, the native BOA is utilized while its continuous steps are bounded in a threshold using a suitable threshold function after squashing them. The proposed binary algorithms are compared with five state-of-the-art approaches and four latest high performing optimization algorithms. A number of assessment indicators are utilized to properly assess and compare the performance of these algorithms over 21 datasets from the UCI repository. The experimental results confirm the efficiency of the proposed approaches in improving the classification accuracy compared to other wrapper-based algorithms, which proves the ability of BOA algorithm in searching the feature space and selecting the most informative attributes for classification tasks.

**Keywords:** Binary butterfly optimization algorithm, Butterfly optimization algorithm, Feature selection, Bio-inspired optimization

\*Corresponding author

Email addresses: [sankalap.arora@gmail.com](mailto:sankalap.arora@gmail.com) (Sankalap Arora), [priyanka.sep3@gmail.com](mailto:priyanka.sep3@gmail.com) (Priyanka Anand)

## 1. Introduction

The fundamental aim of any data pre-processing step in the knowledge discovery process is to prepare the dataset to be used by the data mining algorithms (Han et al., 2011). The most important classes of algorithms in the data mining are classification and clustering. Algorithms of both of these classes work on the dimensions of the dataset to make predictions, however, when the dimensionality of the datasets is increased then the performance of the clustering and classification methods is considerably affected (Bennasar et al., 2015). Moreover, high dimensional datasets have various disadvantages such as large time for model construction, redundant data and degraded performance which makes data analysis very difficult. To resolve this issue, feature selection is utilized as a major preprocessing step with the goal of selecting a subset of features out of the large dataset as well as increase the accuracy of the classification or clustering model leading to the removal of noisy, extraneous and ambiguous data.

The feature selection methods have two main steps: search strategy and subset quality evaluation (Zawbaa et al., 2016). In the first step, the search strategy employs some method to select subsets of features. The latter step includes the use of a classifier to evaluate the quality of the subsets obtained by the search strategy module. The feature selection approaches fall into two categories: filter based methods and wrapper-based methods (Liu & Yu, 2005). The filter-based methods used statistical data dependency techniques to find the subset of features. The wrapper-based methods employ machine learning algorithms to look for a near-optimal solution from an exponential set of feasible solutions. The filter-based methods are independent of the classifier and are relatively fast but these methods do not reflect the relevance of the various dimensions while deciding the subset of the features. These techniques use methods such as information gain (Yang & Pedersen, 1997), principal component analysis (Han et al., 2011), mutual information (Peng et al., 2005) and use measures such as the distance between the dimensions, correlation between the dimensions and consistency, etc. The interaction of the features and the classifier is lacking in these methods which is a major limitation. Another drawback is that these methods cannot work with redundant information. On the other hand, the wrapper-based techniques use classifier during their search for the op-

timal subset by employing the classifier's prediction accuracy. The researchers have found that the wrapper-based methods obtain better results than the filter based methods (Zawbaa et al., 2016; Mafarja et al., 2017). The reason for the better performance of these techniques is the use of a machine learning technique (classifier) in the evaluation module.

In the past two decades, nature-inspired meta-heuristic algorithms are at the forefront due to their powerful and efficient performance in dealing with complex real-world problems (Yang, 2010b). These algorithms have the ability to exploit the useful information of the population in order to find the optimal solutions. Some of these algorithms are elephant herd optimizer (Wang et al., 2016c), moth search algorithm (Wang, 2016), cuckoo search (Yang & Deb, 2009; Wang et al., 2016b), monarch butterfly optimization (Wang et al., 2015, 2016d) and krill herd (Gandomi & Alavi, 2012; Wang et al., 2014b,a). These efficient and robust algorithms are utilized intensively to tackle various diverse optimization problems such as complex design problems (Yang, 2010a; Rizk-Allah et al., 2017), node localization in wireless sensor networks (Arora & Singh, 2017d), fault diagnosis (Yi et al., 2016), economic load dispatch (Rizk-Allah et al., 2018), high performance computing (Wang et al., 2017; Cui et al., 2017), high dimension optimization problems (Wang et al., 2016a; Wu et al., 2016), image matching (Zhang & Wang, 2012) and knapsack problem (Feng & Wang, 2018; Feng et al., 2017, 2018).

Moreover, many researchers tried to implement stochastic methods to solve feature selection problems such as simulated annealing (Jensen & Shen, 2004), tabu search (Hedar et al., 2008), Particle Swarm Optimization (PSO) (Bello et al., 2007), artificial bee colony (Wang et al., 2010) and Genetic Algorithm (GA) (Kabir et al., 2011). An attribute reduction algorithm using record-to-record travel algorithm has been utilized using rough set theory to evaluate the quality of the obtained solution (Mafarja & Abdullah, 2013). A fuzzy-based record-to-record travel algorithm has also been employed for solving rough set attribute reduction problems (Mafarja & Abdullah, 2015). In (Kashef & Nezamabadi-pour, 2015), features are assumed as graph nodes to construct a graph model and ant colony optimization is utilized to select nodes in order to solve feature selection problem. A feature selection method based on artificial bee colony and differential evolution has been proposed in (Zorarpaci &

Özel, 2016) and its validation has been done on fifteen datasets from the UCI repository. The Ant Lion Optimizer (ALO) has been used as a wrapper feature selection model in (Emary et al., 2016a; Emary & Zawbaa, 2018). Grey Wolf Optimizer (GWO) (Emary et al., 2016b), flower pollination algorithm (Zawbaa & Emary, 2018) and Dragonfly Algorithm (DA) (Mafarja et al., 2017) are the recent algorithms that have been successfully employed for solving feature selection problems. Chaos-based Salp Swarm Algorithms (SSA) have been employed as feature selection optimization methods in (Ahmad et al., 2018; Sayed et al., 2018). Moreover, an efficient crossover scheme was also utilized to improve the performance of SSA to solve feature selection problem (Faris et al., 2018). Another wrapper feature selection method based on a hybrid GWO-ALO algorithm was proposed in (Zawbaa et al., 2018) by incorporating GWO's global search ability and ALO's exploitation ability to achieve the real optimum. Recently, Whale Optimization Algorithm (WOA) has been employed as a feature selection algorithm in (Mafarja & Mirjalili, 2018). Additionally, SA has been integrated with WOA to search feature subset space for selecting optimal feature set (Mafarja & Mirjalili, 2017).

However, due to the random nature of the above-mentioned techniques, there is no assurance that these techniques will find the optimal feature subset in feature selection problem. Additionally, the No-Free-Lunch theorem attests that there is no single optimization algorithm which is adequate to provide a solution for all optimization algorithms (Mafarja & Mirjalili, 2017). In other words, it means that the present randomization based feature selection techniques can also suffer from degraded performance while solving some problems which influenced our attempts to analyze the efficiencies of the currently proposed Butterfly Optimization Algorithm (BOA). BOA is a recent meta-heuristic algorithm which is inspired by the food foraging behavior of the butterflies (Arora & Singh, 2018, 2015). BOA has attracted the attention of various researchers due to its potential as an optimization technique for global optimization problems and applied to different applications (Arora & Singh, 2017c,d, 2016, 2017a; Arora et al., 2018). BOA demonstrates better performance in comparison to some other optimization algorithms (Arora & Singh, 2018). This population-based metaheuristic has the ability to avoid local optima stagnation to some extent. It also has good convergence ability towards the optima. To the best of our

knowledge, no one has yet systematically applied BOA to solve feature selection problem. This is the main reason which encouraged us to choose BOA as the foundation of our work.

The comprehensive aim of this paper is to propose new binary versions of butterfly optimization algorithm (bBOA) for wrapper feature selection. The proposed algorithms select the optimal feature subset which decreases the feature subset length and at the same time, increases the classification accuracy. The rest of this paper is organized as follows: Section 2 presents the background of continuous butterfly optimization algorithm and the proposed binary versions of butterfly optimization algorithm (bBOA) are described in Section 3. Section 4 presents the proposed bBOA algorithms for feature selection, while the experimental results are reported and analyzed in Section 5. Finally, the conclusions of the research work and the directions for the future work are drawn in Section 6.

## 2. Preliminaries

### 2.1. Continuous butterfly optimization algorithm

Nature inspired metaheuristic algorithms have gained much attention from various researchers in the past (Yang, 2010b). Butterfly Optimization Algorithm (BOA) belongs to the category of bio-inspired algorithms which is a major sub-category of nature inspired metaheuristic algorithms. BOA is basically inspired by the food foraging behavior of butterflies and these butterflies are used as search agents in order to perform optimization in BOA (Arora & Singh, 2018). Biologically, butterflies have sense receptors which are used to smell/sense the fragrance of food/flowers. These sense receptors are called chemoreceptors which are scattered over the butterfly's body parts (Blair & Launer, 1997). In BOA, it is assumed that a butterfly can produce scent/fragrance with some power/intensity. This scent is related to the fitness of the butterfly which is calculated using the objective function of the problem. This implies that when a butterfly moves from one position to another position in the search space, its fitness will change likewise. The scent which is produced by a butterfly can be felt by the different butterflies present in the neighborhood and an aggregate social learning system is framed. When a butterfly senses fragrance from the

best butterfly in the search space, it makes a stride towards the best butterfly and this stage is named as global search phase of BOA. In the second situation, when a butterfly is not able to detect the scent of some other butterfly in the search space, it will make random strides and this stage is named as local search phase. In BOA, the fragrance is formulated as a function of the physical intensity of stimulus as follows (Arora & Singh, 2018; Stevens, 1975):

$$pf_i = cI^a \quad (1)$$

where  $pf_i$  is the perceived magnitude of fragrance, i.e., how strongly the fragrance of  $i^{th}$  butterfly is perceived by other butterflies present in the region,  $c$  is the sensory modality,  $I$  is the stimulus intensity and  $a$  is the power exponent depended on modality, which accounts for varying degree of absorption. In BOA, an artificial butterfly has a position vector which can be updated using during the optimization process using:

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

where  $x_i^t$  is the solution vector  $x_i$  for  $i^{th}$  butterfly in iteration number  $t$  and  $F_i$  represents fragrance which is utilized by  $x_i^{th}$  butterfly to update its position during the course of iterations. Furthermore, there are two key steps in the algorithm, i.e., global search phase and local search phase. In the global search phase, the butterfly takes a step towards the fittest butterfly/solution  $g^*$  which can be represented as:

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

Here  $g^*$  represents the current best solution found among all the solutions of the current iteration. Perceived fragrance of  $i^{th}$  butterfly is represented by  $pf_i$  and  $r$  is a uniform random number in  $[0, 1]$ . Local search phase can be represented as:

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

where  $x_j^t$  and  $x_k^t$  are  $j^{th}$  and  $k^{th}$  butterflies from the solution space. If  $x_j^t$  and  $x_k^t$  belongs to the same population and  $r$  is a uniform random number in  $[0, 1]$

then Eq. (4) becomes a local random walk. A switch probability  $p$  is employed in BOA to switch between common global search to intensive local search. The pseudo-code of the BOA is given in Algorithm 1.

```

1: Objective function  $f(\mathbf{x})$ ,  $\mathbf{x}=(x_1, x_2, \dots, x_d)$ 
2: Generate a population of  $n$  butterflies  $\mathbf{x}_i= (i=1, 2, \dots, n)$ 
3: Define sensor modality  $c$ , power exponent  $a$  and switch probability  $p$ 
4: while stopping criteria are not met do
5:   for each butterfly  $bf$  in population do
6:     Calculate the fragrance for  $bf$  using Eq. (1)
7:   end for
8:   Find the best  $bf$ 
9:   for each butterfly  $bf$  in population do
10:    Generate a random number  $rand$  from  $[0, 1]$ 
11:    if  $rand < p$  then
12:      Move towards the best butterfly using Eq. (2) and (3)
13:    else
14:      Move randomly using Eq. (2) and (4)
15:    end if
16:    Evaluate the new butterfly
17:    If the new butterfly is better, update it in the population
18:  end for
19:  Update the value of  $c$ 
20:  Find the current global best butterfly
21: end while
22: Output the best solution found.

```

**Algorithm 1:** Pseudo-code for BOA

### 3. The proposed binary butterfly optimization algorithm (bBOA)

BOA is a recently proposed bio-inspired algorithm that mimics the food foraging behavior of butterflies (Arora & Singh, 2018, 2017c). BOA demonstrates competitive results in terms of exploration, exploitation, convergence and local optima avoidance. As mentioned in (Arora & Singh, 2018, 2015, 2017a,b), BOA depicts better performance on most of the unimodal and multimodal benchmark functions. The underlying strength of BOA is its exploitation and high convergence rate which is because of the employed random walk and elitism.

On the contrary, BOA is facilitated with high exploration owing to fragrance attenuation which allows it search the solution space efficiently. These attractive properties motivate researchers to use BOA in other applications such as wrapper-based feature selection.

Basically in the wrapper-based selection method, the classifier is utilized for training and assessed at every single iteration and subsequently, an intelligent optimization algorithm is employed to minimize the number of evaluations. Furthermore, the search space is anticipated to be extremely nonlinear with numerous local minima. Generally, continuous optimization algorithms are employed to find that combinations of features which maximize the performance of classifier and the search agents are utilized in  $d$ -dimensional search space at positions in  $[0, 1]$ . On the other hand, binary optimization algorithms are expected to demonstrate better performance, if properly utilized in a similar manner, because the search space is limited to only two values for each dimension  $\{0, 1\}$ . Moreover, the binary operators are considered to be more straightforward than continuous operators.

In the continuous version of BOA, the butterflies update their positions continuously to any position in the search space (Arora & Singh, 2018). The main motivation for developing a binary version of BOA is that in feature selection problems, the solutions are limited to the binary values  $\{0, 1\}$ . In the current research work, a novel binary butterfly optimization algorithm (bBOA) is proposed for feature selection problem. It can be examined that every butterfly is able to update its position using global search phase or local search phase. In the first phase, a butterfly performs walk around the global best butterfly with a suitable step size whereas, in the other phase, a butterfly performs a random walk. As a result, the same search mechanisms are applied in the binary version of the BOA.

As indicated by (Mirjalili & Hashim, 2012; Rashedi et al., 2010; Kennedy & Eberhart, 1997), the concept is to vary the position of a butterfly with the likelihood of its fragrance. In order to achieve this, a transfer function is required to map the fragrance values to the probability values of butterflies in order to update their positions. In alternative words, a transfer function characterizes the probability of changing a butterfly position vector's elements from 0 to 1 and vice versa. Basically, the binary search space can be assumed as a hypercube in

which the butterflies/search agents of BOA can shift only to the farther corners of the hypercube by flipping different numbers of the bits (Kennedy & Eberhart, 1997). The basic concept in the proposed binary versions of BOA is to update the position of a butterfly with the probability of its fragrance. To achieve this, a transfer function is mandatory which would map the fragrance values to the probability values in order to update the positions of the butterflies. In other words, a transfer function provides the value of probability according to which the position vector will be changed from 0 to 1 and vice versa (Mirjalili et al., 2014). In this research work, two transfer functions namely Sigmoid and V-shaped transfer function are utilized to develop two binary versions which are called bBOA-S and bBOA-V respectively. The following two subsections discuss the bBOA-S and bBOA-V as the two proposed binary versions of BOA.

### *3.1. Binary butterfly optimization algorithm - approach 1 (bBOA-S)*

As already discussed, the new positions of the butterflies from global or local search will have continuous solutions and therefore these continuous values must be transformed into corresponding binary values. This conversion is performed by applying squashing of continuous solutions in each dimension using a Sigmoidal (S-shaped) transfer function (Mirjalili & Hashim, 2012) which will force the butterflies to move in a binary search space. The S-shaped function is a common transfer function as given in Eq. (5) and Fig. 1 (Mirjalili & Lewis, 2013; Emary et al., 2016b).

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

where  $F_i^k$  is the continuous-valued fragrance of  $i^{th}$  butterfly in  $k^{th}$  dimension at iteration  $t$ .

The output from the S-shaped transfer function is still in a continuous manner and henceforth it has to be the threshold to reach the binary-valued one. The S-shape functions map the infinite input smoothly to a finite output. It is worthwhile to mention that the probability of changing the values of position vectors rises as the slope of the transfer function increases. The commonly stochastic threshold is applied as mentioned in Eq. (6) to reach the binary

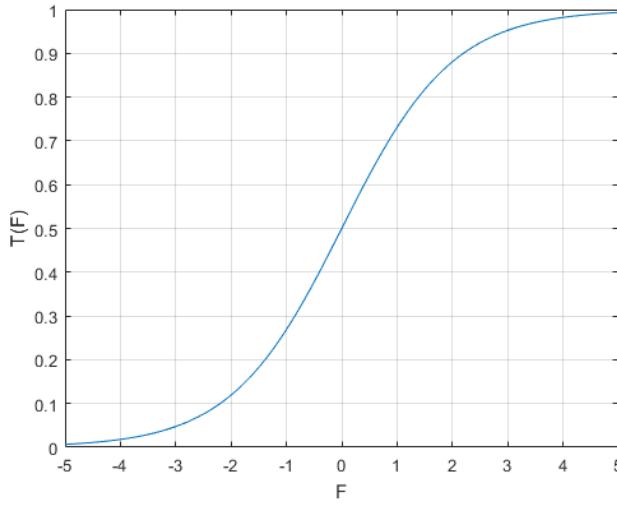


Figure 1: Sigmoid transfer function

solution in case of sigmoidal function.

$$x_i^k(t+1) = \begin{cases} 0 & \text{if } \text{rand} < S(F_i^k(t)) \\ 1 & \text{if } \text{rand} \geq S(F_i^k(t)) \end{cases} \quad (6)$$

where  $x_i^k(t)$  and  $F_i^k(t)$  indicate the position and fragrance of  $i^{th}$  butterfly at iteration  $t$  in  $k^{th}$  dimension.

### 3.2. Binary butterfly optimization algorithm - approach 2 (bBOA-V)

In this approach, instead of a S-shaped transfer function, a V-shaped transfer function is proposed and in order to achieve this Eqs. (7) and (9) are utilized (Mirjalili & Lewis, 2013). The steps of utilizing the proposed transfer function in order to force butterflies to move in a binary search space are illustrated in Fig. 2 (Rashedi et al., 2010; Mirjalili & Lewis, 2013).

$$V(F_i^k(t)) = \left| \operatorname{erf}\left(\frac{\sqrt{\pi}}{2} F_i^k(t)\right) \right| \quad (7)$$

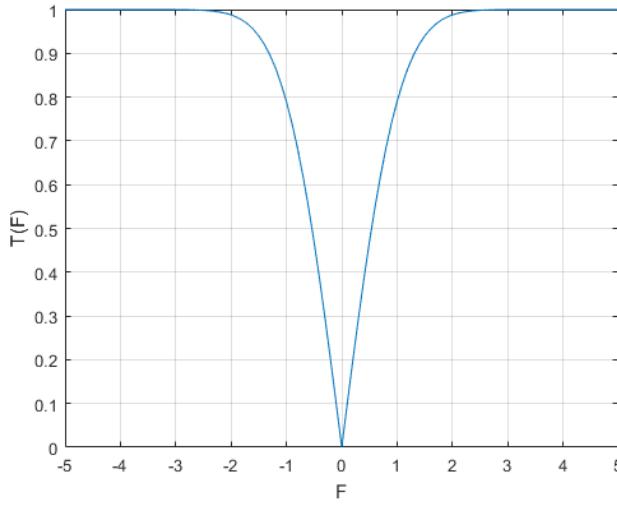


Figure 2: V-shaped transfer function

The Eq. (7) can be rewritten as (Mirjalili & Lewis, 2013):

$$V(F_i^k(t)) = \left| \frac{\sqrt{\pi}}{2} \int_0^{(\sqrt{\pi}/2)F_i^k(t)} e^{-t^2} dt \right| \quad (8)$$

The threshold rules can be represented mathematically as:

$$x_i^k(t+1) = \begin{cases} (x_i^k(t))^{-1} & \text{if } \text{rand} < V(F_i^k(t)) \\ x_i^k(t) & \text{if } \text{rand} \geq V(F_i^k(t)) \end{cases} \quad (9)$$

where  $x_i^k(t)$  and  $F_i^k(t)$  indicate the position and fragrance of  $i^{th}$  butterfly at iteration  $t$  in  $k^{th}$  dimension, and  $x_i^k(t)^{-1}$  is the complement of  $x_i^k(t)$ .

In this binary approach, Eq. (7) is utilized as the transfer function with the aim to transform the fragrances of butterflies to the probabilities of changing their position vectors' elements. Accordingly, the rules of Eq. (9) are employed to update the position vectors of butterflies. The benefit of V-Shaped transfer function is that it does not force butterflies to take 0 or 1 value. In other words, it encourages butterflies to switch to their compliments only when their fragrance values are high otherwise the butterflies will stay in their current positions considering their low fragrance values.

The general steps of binary BOA are presented in Algorithm 2 and the efficiency of binary BOA's are investigated in the following section.

```

1: Objective function  $f(\mathbf{x})$ ,  $\mathbf{x}=(x_1, x_2, \dots, x_d)$ 
2: Generate a population of  $n$  butterflies  $\mathbf{x}_i= (i=1, 2, \dots, n)$ 
3: Define sensor modality  $c$ , power exponent  $a$  and switch probability  $p$ 
4: while stopping criteria are not met do
5:   for each butterfly  $bf$  in population do
6:     Calculate the fragrance for  $bf$  using Eq. (1)
7:   end for
8:   Find the best  $bf$ 
9:   for each butterfly  $bf$  in population do
10:    Generate a random number  $rand$  from  $[0, 1]$ 
11:    if  $rand < p$  then
12:      Move towards the best butterfly using Eq. (2) and (3)
13:    else
14:      Move randomly using Eq. (2) and (4)
15:    end if
16:    Calculate the value of transfer function using Eq. (5) or (7)
17:    Squash the solution using Eq. (6) or (9)
18:    Evaluate the new butterfly
19:    If the new butterfly is better, update it in the population
20:  end for
21:  Update the value of  $c$ 
22:  Find the current global best butterfly
23: end while
24: Output the best solution found.

```

**Algorithm 2:** Pseudo-code for binary BOA

#### 4. Binary butterfly optimization algorithm for feature selection

Feature selection is a binary optimization problem in which the search agents are confined to binary  $\{0, 1\}$  values only. In this research work, every solution is characterized as a single dimensional vector in which the length of the vector depends on the number of features/attributes in the dataset. Every cell of the vector can contain two values, i.e., 1 or 0, where the value 1 depicts that the corresponding feature/attribute is chosen whereas the value 0 represents that the feature/attribute is not selected.

Feature selection problem can be considered as a multi-objective optimization problem in which two opposing goals are to be accomplished; selecting minimum number of features and maximum classification accuracy. In order to solve this multi-objective problem, two binary optimization approaches are proposed. In feature selection problem, that solution is considered best which contains the minimal number of features along with the highest classification accuracy. Every solution is assessed by the proposed fitness function which relies on KNN classifier (Altman, 1992) in order to calculate its classification accuracy and on the number of features selected. Keeping in mind the end goal which is to find the balance between the number of attributes and classification accuracy, the fitness function in Eq. (10) is employed in all the optimization algorithms in order to evaluate the solutions.

$$\text{Fitness} = \alpha\gamma_R(D) + \beta\frac{|R|}{|N|} \quad (10)$$

where  $\gamma_R(D)$  is the classification error rate of the KNN classifier. Furthermore,  $|R|$  represents the cardinality of the selected feature subset and  $|N|$  represents the total number of features in the original dataset,  $\alpha$  and  $\beta$  are two parameters corresponding to the importance of classification quality and subset length,  $\alpha \in [0, 1]$  and  $\beta = (1 - \alpha)$  adopted from literature (Emary et al., 2016a; Mafarja & Mirjalili, 2017).

## 5. Experimental results and discussion

### 5.1. Data description

In order to validate the performance of the proposed binary approaches, twenty one benchmark datasets from the UCI data repository are chosen for experiments (Emary et al., 2016a; Zawbaa et al., 2016). Table 1 depicts the details of the selected datasets such as number of features and instances in every dataset. The underlying reason for selecting these data sets is that these contain a number of attributes and instances which represent a variety of issues on which the proposed binary approaches will be tested on (Emary et al., 2016a). Furthermore, a set of high dimensional data sets is also selected in order to assess the performance of proposed approaches in high search spaces. For evaluation

purposes, every data set is divided in cross-validation manner (Hastie et al., 2001). Fundamentally, in  $K$ -fold cross-validation, the dataset is divided into a number of folds where  $K - 1$  folds are utilized for training and the rest of the fold is utilized for the testing purpose. This procedure is reiterated for  $M$  times. Thus, each optimization algorithm is assessed  $K \times M$  times for each data set. The training, testing and validation data is divided into equal sized partitions. The training fraction is dedicated to the training of the classifier during the optimization process whereas the validation fraction is employed to evaluate the performance of the classifier during the optimization time. The testing fraction is utilized to assess the selected feature set provided by the trained classifier.

In this research work, a wrapper-based approach for feature selection is employed on the basis of the KNN classifier for which the best choice ( $k = 5$ ) is used on all the data sets (Hastie et al., 2001). In the training process, each search agent depicts single feature subset. Throughout the optimization process, in order to guide the optimal feature subset selection process, the training fold is utilized to assess the performance of KNN classifier on the validation fold whereas the test fold is kept hidden from the optimization process and is utilized for final assessment purpose only.

In order to benchmark the proposed feature selection approaches, their performances are compared with various the state-of-the-art approaches; GA (Holland & Goldberg, 1989), PSO (Eberhart & Kennedy, 1995), ALO (Emary et al., 2016a), Brain Storm Optimizer (BSO) (Shi, 2011) and GWO (Emary et al., 2016b). Furthermore, the proposed feature selection approaches are also compared with latest optimization algorithms; DA (Mirjalili, 2016a), Sine Cosine Algorithm (SCA) (Mirjalili, 2016b), SSA (Mirjalili et al., 2017), WOA (Mafarja & Mirjalili, 2017, 2018). The global and algorithm-specific parameter settings are mentioned in Table 2. The proposed approaches are implemented using MATLAB R2009b (The MathWorks, Inc., U.S., Version 7.9.0.529), under Microsoft Windows 8 operating system. All simulations are carried out on a CPU i5 – 3210 (Intel CoreTM Processor @2.50 GHz) computer.

Table 1: List of data sets used in the experiments

S. No.	Name	No. of features	No. of samples
1	Breastcancer	9	699
2	BreastEW	30	569
3	Clean1	166	476
4	Clean2	166	6598
5	CongressEW	16	435
6	Exactly	13	1000
7	Exactly2	13	1000
8	HeartEW	13	270
9	IonosphereEW	34	351
10	KrvskpEW	36	3196
11	Lymphography	18	148
12	M-of-n	13	1000
13	PenglungEW	325	73
14	Semeion	265	1593
15	SonarEW	60	208
16	SpectEW	22	267
17	Tic-tac-toe	9	958
18	Vote	16	300
19	WaveformEW	40	5000
20	WineEW	13	178
21	Zoo	16	101

### 5.2. Evaluation criteria

The following measures are utilized on data in every run of individual optimization algorithm:

- Classification accuracy: is an indicator which describes how accurate is the classifier given the selected feature set, i.e, the number of features correctly classified and the classification accuracy in this study is calculated as:

$$\text{Average performance} = \frac{1}{M} \sum_{j=1}^M \frac{1}{N} \sum_{i=1}^N \text{match}(C_i, L_i), \quad (11)$$

where  $M$  is the number of times the optimization algorithm has been run,  $N$  denotes the number of test set points,  $C_i$  is output class label for

Table 2: Parameter setting for experiments.

Parameter	Value (s)
K for cross validation	5
M number of runs	20
No. of search agents	7
No. of iterations	100
Problem dimension	No. of features in the data set
Search domain	{0, 1}
Crossover ratio in GA	0.9
Mutation ratio in GA	0.1
Selection mechanism in GA	Roulette wheel
Acceleration constants in PSO	[2, 2]
Inertia w in PSO	[0.9, 0.6]
$A$ parameter in GWO	Min = 0 and max = 2
Mutation rate $r$ in ALO	Min = 0 and max = 0.9
No. of clusters in BSO	5
Probability in BSO	$P_{5a}=0.2, P_{6b}=0.8, P_{6bi}=0.4, P_{6c}=0.5$
$r4$ parameter in SCA	0.5
$A$ parameter in SCA	2
$A$ parameter in WOA	Min = 0 and max = 2
$a$ parameter in BOA/bBOA	0.1
$c$ parameter in BOA/bBOA	Min = 0.01 and max = 0.25

particular data point  $i$ ,  $L_i$  is reference class label for  $i$ , and  $match$  is the comparison function that gives outputs 1 when two labels are the same and outputs 0 when they are different.

- Average selection size: depicts the mean of the selected features for  $M$  times and can be represented as:

$$Mean = \frac{1}{M} \sum_{i=1}^M size(g^*) \quad (12)$$

where  $\text{size}(x)$  represents the number of features selected in the testing data set.

- Statistical mean: represents the average of solutions calculated by executing an optimization algorithm for  $M$  times and can be formulated as:

$$\text{Mean} = \frac{1}{M} \sum_{i=1}^M g^{*i} \quad (13)$$

where  $g^{*i}$  represents the best solution in the  $i^{th}$  run.

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

$$\text{Std. dev.} = \sqrt{\frac{1}{M-1} \sum (g^{*i} - \text{Mean})^2} \quad (14)$$

where  $g^{*i}$  represents the best solution in the  $i^{th}$  run.

- Statistical best: represents the best solution obtained by executing an optimization algorithm over  $M$  times.
- Statistical worst: represents the worst solution obtained by executing an optimization algorithm over  $M$  times.
- Average computational time: is the actual runtime of an individual optimization algorithm in milliseconds over different runs and it can be calculated using:

$$\text{Mean} = \frac{1}{M} \sum_{i=1}^M \text{RunTime}_{o,i} \quad (15)$$

where  $M$  represents the number of runs for an optimization algorithm  $o$ , and  $\text{RunTime}_{o,i}$  represents the actual computational time for the optimization algorithm  $o$  at run number  $i$ .

- Wilcoxon rank sum test: is a nonparametric statistical test used to check whether the results of proposed approaches are statistically different from other algorithms (Derrac et al., 2011). In this test, ranks are assigned to all the values considered as a single group and afterwards ranks of every

group are added. The null hypothesis is that the two samples are from the same population, therefore any difference in the two rank sums come only from sampling error. This statistical test returns a parameter called *p*-value which is utilized to verify the significance level of two algorithms.

### 5.3. Numerical results and discussion

#### 5.3.1. Comparison of BOA, bBOA-S and bBOA-V

Table 3 outlines the results of the proposed approaches in terms of classification accuracy. It is evident that the proposed S-bBOA algorithm performed much better than the native one for classification accuracy. The native BOA does not perform better than any binary BOA over all the datasets used in this study. On the other hand, the performance of V-bBOA is also better than native BOA on all the datasets. It is worthwhile to mention that although S-bBOA outperformed V-bBOA on most of the datasets utilized in this study.

In terms of average selection size, again S-bBOA outperformed the native BOA on all the datasets and the performance of V-bBOA is also competitive with S-bBOA and the results are shown in Table 4. The V-bBOA performed better than S-bBOA on 4 datasets and in the clean2 dataset, V-bBOA outperformed S-bBOA by providing 74.60 average selection size, at the same time S-bBOA demonstrates 92.40 average selection size.

A similar pattern can be seen in the results of S-bBOA on statistical fitness measure (best, mean and worst) which are shown in Tables 5-7. It is evident that S-bBOA outperformed BOA and V-bBOA on best, mean and worst fitness values on all the datasets. On the contrary, on the statistical standard deviation measure, the performance of BOA is competitive to the performance of S-bBOA whereas V-bBOA demonstrates inferior performance among the three optimizers as per the results presented in Table 8. The average computational time of BOAs' approaches is depicted in Table 9 which is the time required to reach the near optimal solution. As it is can be observed from Table 9 that S-bBOA requires the least computational time than BOA on all the datasets whereas V-bBOA performed better than BOA in case of waveformEW dataset only. This can be interpreted easily since BOA is performing its operations as a continuous optimization algorithm which consumes more computational time. On the other

hand, the V-shaped approach requires much more computational time than the S-shaped approach.

Table 3: Comparison between the proposed approaches based on classification accuracy

S. No	Dataset	BOA	S-bBOA	V-bBOA
1	Breastcancer	0.9640	<b>0.9686</b>	0.9663
2	BreastEW	0.9506	<b>0.9709</b>	0.9614
3	Clean1	0.8709	0.8832	<b>0.8916</b>
4	Clean2	0.9474	<b>0.9614</b>	0.9539
5	CongressEW	0.9459	<b>0.9593</b>	0.9587
6	Exactly	0.7676	<b>0.9724</b>	0.9656
7	Exactly2	0.7189	<b>0.7596</b>	0.7408
8	HeartEW	0.8010	<b>0.8237</b>	0.8222
9	IonosphereEW	0.9023	0.9070	<b>0.9295</b>
10	KrvskpEW	0.9328	<b>0.9660</b>	0.9519
11	Lymphography	0.8224	<b>0.8676</b>	0.8641
12	M-of-n	0.8511	<b>0.9720</b>	0.9690
13	PenglungEW	0.8665	0.8775	<b>0.8919</b>
14	Semeion	0.9787	<b>0.9819</b>	0.9789
15	SonarEW	0.8853	<b>0.9362</b>	0.9288
16	SpectEW	0.8154	<b>0.8463</b>	0.8433
17	Tic-tac-toe	0.7744	<b>0.7983</b>	0.7952
18	Vote	0.9369	<b>0.9653</b>	0.9587
19	WaveformEW	0.7285	<b>0.7429</b>	0.7388
20	WineEW	0.9715	<b>0.9843</b>	0.9820
21	Zoo	0.9608	<b>0.9775</b>	0.9725
<b>Average</b>		0.8768	<b>0.9105</b>	0.9078

Table 4: Comparison between the proposed approaches based on average number of features

S. No	Dataset	BOA	S-bBOA	V-bBOA
1	Breastcancer	6.20	<b>5.60</b>	5.80
2	BreastEW	21.07	16.80	<b>12.80</b>
3	Clean1	110.93	<b>91.80</b>	92.00
4	Clean2	93.00	92.40	<b>74.60</b>
5	CongressEW	9.40	<b>6.40</b>	6.50
6	Exactly	10.87	<b>7.60</b>	8.20
7	Exactly2	7.87	<b>4.80</b>	5.40
8	HeartEW	9.13	<b>5.80</b>	6.40
9	IonosphereEW	22.27	<b>16.20</b>	19.20
10	KrvskpEW	24.00	<b>17.60</b>	22.20
11	Lymphography	13.27	<b>8.40</b>	9.20
12	M-of-n	11.07	<b>6.80</b>	7.00
13	PenglungEW	170.80	172.00	<b>164.80</b>
14	Semeion	211.40	<b>143.20</b>	174.40
15	SonarEW	41.07	<b>32.80</b>	33.40
16	SpectEW	15.20	10.80	<b>10.00</b>
17	Tic-tac-toe	7.87	<b>5.60</b>	5.80
18	Vote	9.60	<b>5.20</b>	6.60
19	WaveformEW	30.75	<b>25.00</b>	25.40
20	WineEW	10.27	<b>6.20</b>	6.60
21	Zoo	10.47	<b>5.20</b>	6.00

Table 5: Comparison between the proposed approaches based on mean fitness measure

S. No	Dataset	BOA	S-bBOA	V-bBOA
1	Breastcancer	0.043	<b>0.040</b>	0.048
2	BreastEW	0.056	<b>0.042</b>	0.053
3	Clean1	0.135	<b>0.113</b>	0.121
4	Clean2	0.058	<b>0.051</b>	0.053
5	CongressEW	0.059	<b>0.045</b>	0.051
6	Exactly	0.238	<b>0.040</b>	0.072
7	Exactly2	0.284	<b>0.260</b>	0.272
8	HeartEW	0.204	<b>0.180</b>	0.182
9	IonosphereEW	0.103	<b>0.096</b>	0.099
10	KrvskpEW	0.073	<b>0.054</b>	0.079
11	Lymphography	0.183	<b>0.139</b>	0.146
12	M-of-n	0.156	<b>0.027</b>	0.036
13	PenglungEW	0.128	<b>0.118</b>	0.164
14	Semeion	0.029	<b>0.026</b>	0.028
15	SonarEW	0.120	<b>0.086</b>	0.108
16	SpectEW	0.190	<b>0.160</b>	0.167
17	Tic-tac-toe	0.232	<b>0.205</b>	0.211
18	Vote	0.068	<b>0.044</b>	0.048
19	WaveformEW	0.276	<b>0.265</b>	0.271
20	WineEW	0.036	<b>0.023</b>	0.025
21	Zoo	0.047	<b>0.034</b>	0.037

Table 6: Comparison between the proposed approaches based on best fitness measure

S. No	Dataset	BOA	S-bBOA	V-bBOA
1	Breastcancer	0.039	<b>0.024</b>	0.032
2	BreastEW	0.040	<b>0.032</b>	0.046
3	Clean1	0.092	<b>0.088</b>	0.093
4	Clean2	0.055	<b>0.037</b>	0.050
5	CongressEW	0.037	<b>0.030</b>	0.038
6	Exactly	0.128	<b>0.005</b>	0.006
7	Exactly2	0.238	<b>0.225</b>	0.239
8	HeartEW	0.159	<b>0.138</b>	0.166
9	IonosphereEW	0.077	<b>0.060</b>	0.078
10	KrvskpEW	0.066	<b>0.036</b>	0.042
11	Lymphography	0.156	<b>0.073</b>	0.099
12	M-of-n	0.127	<b>0.005</b>	<b>0.005</b>
13	PenglungEW	0.058	<b>0.033</b>	0.085
14	Semeion	0.033	<b>0.029</b>	0.031
15	SonarEW	0.111	<b>0.072</b>	0.072
16	SpectEW	0.155	<b>0.122</b>	0.130
17	Tic-tac-toe	0.215	<b>0.195</b>	0.197
18	Vote	0.051	<b>0.016</b>	0.030
19	WaveformEW	0.265	<b>0.254</b>	0.265
20	WineEW	0.021	<b>0.005</b>	0.006
21	Zoo	0.004	<b>0.002</b>	0.006

Table 7: Comparison between the proposed approaches based on worst fitness measure

S. No	Dataset	BOA	S-bBOA	V-bBOA
1	Breastcancer	0.046	<b>0.041</b>	0.047
2	BreastEW	0.066	<b>0.049</b>	0.060
3	Clean1	0.149	<b>0.138</b>	0.139
4	Clean2	0.062	<b>0.048</b>	0.053
5	CongressEW	0.075	<b>0.058</b>	0.085
6	Exactly	0.269	<b>0.005</b>	0.340
7	Exactly2	0.313	<b>0.291</b>	0.318
8	HeartEW	0.221	<b>0.195</b>	0.201
9	IonosphereEW	0.173	<b>0.118</b>	0.120
10	KrvskpEW	0.086	<b>0.064</b>	0.141
11	Lymphography	0.211	<b>0.099</b>	0.207
12	M-of-n	0.178	<b>0.110</b>	0.161
13	PenglungEW	0.272	<b>0.169</b>	0.220
14	Semeion	0.033	<b>0.029</b>	0.031
15	SonarEW	0.143	<b>0.109</b>	0.158
16	SpectEW	0.218	<b>0.209</b>	0.210
17	Tic-tac-toe	0.248	<b>0.216</b>	0.227
18	Vote	0.080	<b>0.057</b>	0.058
19	WaveformEW	0.281	<b>0.265</b>	0.269
20	WineEW	0.043	<b>0.028</b>	0.037
21	Zoo	0.120	<b>0.048</b>	0.082

Table 8: Comparison between the proposed approaches based on standard deviation measure

S. No	Dataset	BOA	S-bBOA	V-bBOA
1	Breastcancer	0.006	0.006	<b>0.005</b>
2	BreastEW	<b>0.002</b>	0.003	0.008
3	Clean1	0.011	<b>0.010</b>	0.018
4	Clean2	0.040	<b>0.010</b>	0.150
5	CongressEW	0.024	<b>0.020</b>	0.034
6	Exactly	0.022	<b>0.011</b>	0.018
7	Exactly2	<b>0.011</b>	0.059	0.020
8	HeartEW	0.009	<b>0.008</b>	0.049
9	IonosphereEW	0.017	<b>0.010</b>	0.043
10	KrvskpEW	0.019	<b>0.003</b>	0.070
11	Lymphography	<b>0.009</b>	0.018	0.041
12	M-of-n	<b>0.016</b>	0.035	0.036
13	PenglungEW	0.011	<b>0.007</b>	0.013
14	Semeion	<b>0.009</b>	0.010	0.015
15	SonarEW	0.008	<b>0.001</b>	0.007
16	SpectEW	<b>0.006</b>	0.010	0.012
17	Tic-tac-toe	<b>0.015</b>	0.044	0.033
18	Vote	<b>0.015</b>	0.016	0.018
19	WaveformEW	0.003	<b>0.001</b>	0.004
20	WineEW	<b>0.036</b>	0.056	0.096
21	Zoo	<b>0.003</b>	0.004	0.003

Table 9: Comparison between the proposed approaches based on average computational time

S. No	Dataset	BOA	S-bBOA	V-bBOA
1	Breastcancer	4.41	<b>2.36</b>	5.19
2	BreastEW	4.53	<b>2.41</b>	5.47
3	Clean1	6.13	<b>3.75</b>	7.70
4	Clean2	709.90	<b>236.46</b>	649.28
5	CongressEW	4.80	<b>2.63</b>	5.09
6	Exactly	7.92	<b>4.87</b>	6.73
7	Exactly2	7.44	<b>4.11</b>	6.95
8	HeartEW	4.42	<b>2.41</b>	4.52
9	IonosphereEW	4.89	<b>2.61</b>	4.57
10	KrvskpEW	32.18	<b>15.16</b>	22.04
11	Lymphography	4.36	<b>2.78</b>	4.40
12	M-of-n	7.13	<b>4.28</b>	6.74
13	PenglungEW	4.84	<b>2.54</b>	4.78
14	Semeion	47.42	<b>23.34</b>	29.17
15	SonarEW	4.49	<b>2.54</b>	4.57
16	SpectEW	4.44	<b>2.38</b>	4.57
17	Tic-tac-toe	7.88	<b>4.54</b>	4.93
18	Vote	4.56	<b>2.53</b>	3.65
19	WaveformEW	77.90	89.42	<b>47.57</b>
20	WineEW	4.48	<b>2.43</b>	3.65
21	Zoo	4.63	<b>2.24</b>	3.79

### 5.3.2. Comparison with the state-of-the-art approaches

In the last section, it can be analyzed that S-bBOA has demonstrated superior performance in comparison to other proposed approaches in terms of classification accuracy as well as average selection size. In the current section, the performance of the best approach, i.e., S-bBOA is compared with various state-of-the-art approaches which are widely applied to solve the feature selection problem in the literature. Table 10 outlines the results of S-bBOA, ALO, BSO, GA, GWO and PSO in terms of classification accuracy.

As per the results presented in Table 10, S-bBOA outperforms all other approaches on all the datasets which clearly demonstrates the strength of the proposed approach. PSO is in second place and GWO is in third place. In Table 11, the average of selected attributes using S-bBOA and other approaches are reported. S-bBOA demonstrates much better performance than the other approaches on all the datasets. The underlying reason behind this performance is the enhanced exploration and exploitation capability of S-bBOA which intensively searches the high-performance regions of the feature space.

The statistical measures (mean, best, worst and standard deviation) obtained on the different runs of the algorithms on all the data sets are presented in Tables 12-15. It is evident from these tables that S-bBOA outperforms ALO, BSO, GA, GWO and PSO in mean, best and worst fitness criteria on all the datasets whereas in standard deviation fitness criteria, S-bBOA has shown competitive performance in comparison to other approaches and it is not worse than any other algorithm on seven datasets. In Table 16, the average computational time of S-bBOA is compared with the state-of-the-art approaches. All the optimization algorithms are utilizing a similar number of iterations and therefore computational time has been used as a comparison factor for the performance of the algorithms. In Table 16, GA has the best computational time on eight datasets whereas S-bBOA has performed second by performing better on seven datasets and GWO outperformed other optimizers on six datasets. It can be analyzed that the proposed S-bBOA has demonstrated competitive computational speed in comparison to the state-of-the-art approaches. Therefore, it can be stated that the S-bBOA performed well overall in comparison to the state-of-the-art approaches.

Table 10: Comparison between the proposed S-bBOA and state-of-the-art approaches based on the classification accuracy

S. No.	Dataset	<b>ALO</b>	<b>BSO</b>	<b>GA</b>	<b>GWO</b>	<b>PSO</b>	<b>S-bBOA</b>
1	Breastcancer	0.9591	0.9200	0.9597	0.9603	0.9609	<b>0.9686</b>
2	BreastEW	0.9392	0.9020	0.9488	0.9375	0.9378	<b>0.9709</b>
3	Clean1	0.8465	0.8261	0.8697	0.8580	0.8549	<b>0.8832</b>
4	Clean2	0.9496	0.9391	0.9423	0.9463	0.9465	<b>0.9614</b>
5	CongressEW	0.9370	0.8547	0.9413	0.9327	0.9235	<b>0.9593</b>
6	Exactly	0.7061	0.6021	0.7306	0.7249	0.7471	<b>0.9724</b>
7	Exactly2	0.6980	0.6345	0.6940	0.6929	0.6959	<b>0.7596</b>
8	HeartEW	0.7773	0.6948	0.7867	0.7768	0.7788	<b>0.8237</b>
9	IonosphereEW	0.8595	0.8538	0.8938	0.8682	0.8803	<b>0.9070</b>
10	KrvskpEW	0.9006	0.7603	0.9215	0.9143	0.9200	<b>0.9660</b>
11	Lymphography	0.7863	0.6931	0.8164	0.7629	0.7906	<b>0.8676</b>
12	M-of-n	0.8184	0.7033	0.7988	0.8272	0.8425	<b>0.9720</b>
13	PenglungEW	0.8072	0.7676	0.6721	0.8341	0.8140	<b>0.8775</b>
14	Semeion	0.9714	0.9604	0.9757	0.9671	0.9676	<b>0.9819</b>
15	SonarEW	0.8487	0.7936	0.8750	0.8622	0.8667	<b>0.9362</b>
16	SpectEW	0.7881	0.7507	0.8097	0.7846	0.7841	<b>0.8463</b>
17	Tic-tac-toe	0.7587	0.6601	0.7609	0.7537	0.7518	<b>0.7983</b>
18	Vote	0.9258	0.8413	0.9333	0.9196	0.9258	<b>0.9653</b>
19	WaveformEW	0.7066	0.6150	0.6921	0.7096	0.7192	<b>0.7429</b>
20	WineEW	0.9543	0.8652	0.9536	0.9476	0.9521	<b>0.9843</b>
21	Zoo	0.9216	0.8131	0.9294	0.9525	0.9451	<b>0.9775</b>
<b>Average</b>		0.8505	0.7834	0.8526	0.8540	0.8574	<b>0.9105</b>

Table 11: Comparison between the proposed S-bBOA and state-of-the-art approaches based on average number of features

S. No	Dataset	ALO	BSO	GA	GWO	PSQ	S-bBOA
1	Breastcancer	7.00	6.40	6.10	6.90	5.70	<b>5.60</b>
2	BreastEW	24.27	13.73	<b>12.20</b>	19.00	18.33	16.80
3	Clean1	132.00	98.73	98.90	109.60	104.93	<b>91.80</b>
4	Clean2	95.00	101.00	94.10	106.00	109.40	<b>92.40</b>
5	CongressEW	9.87	7.53	7.10	9.80	10.80	<b>6.40</b>
6	Exactly	12.87	7.73	8.10	12.07	9.00	<b>7.60</b>
7	Exactly2	8.40	6.27	7.10	7.53	9.40	<b>4.80</b>
8	HeartEW	10.40	6.60	<b>6.60</b>	8.80	9.07	<b>5.80</b>
9	IonosphereEW	20.13	15.93	<b>13.50</b>	17.33	19.20	16.20
10	KrvskpEW	35.80	17.80	18.00	31.60	25.60	<b>17.60</b>
11	Lymphography	13.33	9.47	8.90	11.80	11.73	<b>8.40</b>
12	M-of-n	11.27	6.90	7.68	11.27	10.87	<b>6.80</b>
13	PenglungEW	172.07	160.60	<b>153.00</b>	162.80	183.33	172.00
14	Semeion	187.80	162.00	149.40	203.60	171.60	<b>143.20</b>
15	SonarEW	48.00	30.60	<b>30.30</b>	41.60	37.60	32.80
16	SpectEW	13.87	10.87	<b>7.00</b>	13.20	12.07	10.80
17	Tic-tac-toe	8.80	5.88	5.80	7.53	6.73	<b>5.60</b>
18	Vote	8.40	7.87	5.80	8.47	9.33	<b>5.20</b>
19	WaveformEW	39.60	29.00	30.40	36.60	35.80	<b>25.00</b>
20	WineEW	11.07	6.67	6.73	10.73	10.07	<b>6.20</b>
21	Zoo	11.67	7.67	5.35	12.40	11.80	<b>5.20</b>

Table 12: Comparison between the proposed S-bBOA and state-of-the-art approaches based on mean fitness measure

S. No	Dataset	ALO	BSO	GA	GWO	PSO	S-bBOA
1	Breastcancer	0.048	0.084	0.046	0.047	0.045	<b>0.040</b>
2	BreastEW	0.068	0.102	0.055	0.068	0.068	<b>0.042</b>
3	Clean1	0.160	0.177	0.134	0.147	0.150	<b>0.113</b>
4	Clean2	0.055	0.065	0.062	0.060	0.060	<b>0.051</b>
5	CongressEW	0.069	0.149	0.063	0.073	0.082	<b>0.045</b>
6	Exactly	0.301	0.400	0.270	0.282	0.257	<b>0.040</b>
7	Exactly2	0.305	0.367	0.308	0.310	0.308	<b>0.260</b>
8	HeartEW	0.228	0.307	0.216	0.228	0.226	<b>0.180</b>
9	IonosphereEW	0.145	0.149	0.109	0.136	0.124	<b>0.096</b>
10	KrvskpEW	0.108	0.242	0.083	0.094	0.086	<b>0.054</b>
11	Lymphography	0.219	0.309	0.187	0.241	0.214	<b>0.139</b>
12	M-of-n	0.188	0.299	0.205	0.180	0.164	<b>0.027</b>
13	PenglungEW	0.196	0.235	0.129	0.169	0.190	<b>0.118</b>
14	Semeion	0.035	0.044	0.029	0.040	0.039	<b>0.026</b>
15	SonarEW	0.158	0.209	0.128	0.143	0.138	<b>0.086</b>
16	SpectEW	0.216	0.252	0.192	0.219	0.219	<b>0.160</b>
17	Tic-tac-toe	0.249	0.342	0.243	0.252	0.253	<b>0.205</b>
18	Vote	0.079	0.162	0.070	0.085	0.079	<b>0.044</b>
19	WaveformEW	0.300	0.386	0.310	0.297	0.287	<b>0.265</b>
20	WineEW	0.054	0.139	0.051	0.060	0.055	<b>0.023</b>
21	Zoo	0.085	0.190	0.073	0.055	0.062	<b>0.034</b>

Table 13: Comparison between the proposed S-bBOA and state-of-the-art approaches based on best fitness measure

S. No	Dataset	ALO	BSO	GA	GWO	PSO	S-bBOA
1	Breastcancer	0.038	0.046	0.040	0.038	0.039	<b>0.024</b>
2	BreastEW	0.056	0.065	0.048	0.048	0.049	<b>0.032</b>
3	Clean1	0.118	0.130	0.122	0.117	0.100	<b>0.088</b>
4	Clean2	0.049	0.058	0.062	0.056	0.058	<b>0.037</b>
5	CongressEW	0.044	0.076	0.054	0.042	0.048	<b>0.030</b>
6	Exactly	0.267	0.328	0.015	0.173	0.138	<b>0.005</b>
7	Exactly2	0.252	0.296	0.295	0.279	0.275	<b>0.225</b>
8	HeartEW	0.172	0.206	0.202	0.189	0.178	<b>0.138</b>
9	IonosphereEW	0.111	0.101	0.099	0.088	0.081	<b>0.060</b>
10	KrvskpEW	0.093	0.133	0.063	0.090	0.052	<b>0.036</b>
11	Lymphography	0.165	0.220	0.168	0.193	0.179	<b>0.073</b>
12	M-of-n	0.160	0.170	0.140	0.128	0.064	<b>0.005</b>
13	PenglungEW	0.085	0.085	0.137	0.085	0.086	<b>0.033</b>
14	Semeion	0.041	0.046	0.033	0.044	0.042	<b>0.029</b>
15	SonarEW	0.128	0.139	0.109	0.090	0.091	<b>0.072</b>
16	SpectEW	0.144	0.198	0.170	0.149	0.166	<b>0.122</b>
17	Tic-tac-toe	0.213	0.252	0.232	0.223	0.204	<b>0.195</b>
18	Vote	0.043	0.065	0.061	0.051	0.039	<b>0.016</b>
19	WaveformEW	0.294	0.338	0.312	0.283	0.271	<b>0.254</b>
20	WineEW	0.029	0.061	0.038	0.019	0.028	<b>0.005</b>
21	Zoo	0.026	0.025	0.061	0.007	0.008	<b>0.002</b>

Table 14: Comparison between the proposed S-bBOA and state-of-the-art approaches based on worst fitness measure

S. No	Dataset	ALO	BSO	GA	GWO	PSO	S-bBOA
1	Breastcancer	0.059	0.196	0.051	0.054	0.060	<b>0.041</b>
2	BreastEW	0.083	0.144	0.063	0.085	0.078	<b>0.049</b>
3	Clean1	0.193	0.208	0.143	0.187	0.186	<b>0.138</b>
4	Clean2	0.060	0.073	0.071	0.063	0.061	<b>0.048</b>
5	CongressEW	0.110	0.267	0.083	0.110	0.149	<b>0.058</b>
6	Exactly	0.343	0.448	0.378	0.344	0.384	<b>0.005</b>
7	Exactly2	0.355	0.517	0.331	0.333	0.335	<b>0.291</b>
8	HeartEW	0.289	0.378	0.261	0.256	0.288	<b>0.195</b>
9	IonosphereEW	0.168	0.195	0.134	0.179	0.163	<b>0.118</b>
10	KrvskpEW	0.118	0.344	0.150	0.096	0.164	<b>0.064</b>
11	Lymphography	0.251	0.378	0.220	0.299	0.276	<b>0.099</b>
12	M-of-n	0.224	0.391	0.288	0.235	0.287	<b>0.110</b>
13	PenglungEW	0.300	0.460	0.190	0.246	0.328	<b>0.169</b>
14	Semeion	0.041	0.046	0.033	0.044	0.042	<b>0.029</b>
15	SonarEW	0.216	0.253	0.156	0.218	0.187	<b>0.109</b>
16	SpectEW	0.271	0.322	0.218	0.265	0.265	<b>0.209</b>
17	Tic-tac-toe	0.275	0.436	0.255	0.309	0.331	<b>0.216</b>
18	Vote	0.113	0.256	0.088	0.169	0.119	<b>0.057</b>
19	WaveformEW	0.304	0.434	0.319	0.316	0.303	<b>0.265</b>
20	WineEW	0.073	0.303	0.082	0.142	0.075	<b>0.028</b>
21	Zoo	0.158	0.430	0.101	0.203	0.182	<b>0.048</b>

Table 15: Comparison between the proposed S-bBOA and state-of-the-art approaches based on standard deviation fitness measure

S. No	Dataset	ALO	BSO	GA	GWO	PSO	S-bBOA
1	Breastcancer	0.008	0.023	<b>0.005</b>	0.011	0.008	0.006
2	BreastEW	0.007	0.044	<b>0.003</b>	0.006	0.006	0.003
3	Clean1	0.017	0.048	<b>0.008</b>	0.019	0.026	0.01
4	Clean2	0.023	0.036	0.136	<b>0.051</b>	0.067	<b>0.01</b>
5	CongressEW	0.03	0.056	<b>0.011</b>	0.013	0.019	0.02
6	Exactly	0.031	0.051	0.02	0.023	0.029	<b>0.011</b>
7	Exactly2	0.018	0.023	<b>0.012</b>	0.022	0.023	0.059
8	HeartEW	0.009	0.094	0.038	<b>0.002</b>	0.045	0.008
9	IonosphereEW	0.026	0.049	0.016	0.03	0.028	<b>0.01</b>
10	KrvskpEW	0.02	0.074	0.054	0.03	0.058	<b>0.003</b>
11	Lymphography	0.025	0.037	<b>0.013</b>	0.04	0.029	0.018
12	M-of-n	0.035	0.036	<b>0.016</b>	0.031	0.027	0.035
13	PenglungEW	0.02	0.053	<b>0.006</b>	0.026	0.034	0.007
14	Semeion	0.018	0.057	<b>0.008</b>	0.029	0.019	0.01
15	SonarEW	0.004	0.038	0.003	0.012	0.012	<b>0.001</b>
16	SpectEW	0.012	0.067	0.011	0.033	0.012	<b>0.01</b>
17	Tic-tac-toe	0.035	0.128	<b>0.015</b>	0.047	0.05	0.044
18	Vote	0.021	0.026	<b>0.009</b>	0.022	0.028	0.016
19	WaveformEW	0.004	0.005	0.004	0.002	0.001	<b>0.001</b>
20	WineEW	0.072	0.102	<b>0.018</b>	0.046	0.077	0.056
21	Zoo	0.007	0.002	<b>0.001</b>	0.005	0.002	0.004

Table 16: Comparison between the proposed S-bBOA and state-of-the-art approaches based on the average running time

S. No	Dataset	ALO	BSO	GA	GWO	PSO	S-bBOA
1	Breastcancer	4.9	3.02	<b>2.34</b>	3.45	2.39	2.36
2	BreastEW	2.87	2.85	<b>2.37</b>	3.61	2.43	2.41
3	Clean1	5.31	3.58	4.07	<b>3.39</b>	3.66	3.75
4	Clean2	610.83	223.69	279.32	<b>158.67</b>	227.16	236.46
5	CongressEW	2.88	3.33	2.81	3.32	2.64	<b>2.63</b>
6	Exactly	3.92	4.58	<b>3.85</b>	4.04	4.53	4.87
7	Exactly2	4.22	4.62	4.15	4.52	4.82	<b>4.11</b>
8	HeartEW	2.69	2.96	2.46	3.09	2.49	<b>2.41</b>
9	IonosphereEW	3.14	3.1	<b>2.58</b>	3.25	2.64	2.61
10	KrvskpEW	18.16	11.56	10.42	<b>9.53</b>	17.16	15.16
11	Lymphography	2.68	2.94	<b>2.46</b>	2.98	3.04	2.78
12	M-of-n	4.08	4.07	3.53	<b>3.39</b>	4.56	4.28
13	PenglungEW	7.65	3.1	2.51	<b>2.49</b>	2.56	2.54
14	Semeion	28.41	14.33	<b>13.1</b>	31.67	24.51	23.34
15	SonarEW	3.3	2.93	<b>2.39</b>	2.97	2.62	2.54
16	SpectEW	2.88	2.96	2.45	3	2.4	<b>2.38</b>
17	Tic-tac-toe	4.36	3.99	<b>3.28</b>	4.38	4.49	4.54
18	Vote	2.89	3.26	2.62	3.25	2.57	<b>2.53</b>
19	WaveformEW	40.51	25.03	23.48	<b>20.63</b>	90.09	89.42
20	WineEW	2.68	2.92	2.46	3.13	2.45	<b>2.43</b>
21	Zoo	2.79	4.85	2.33	3.25	2.24	<b>2.24</b>

### 5.3.3. Comparison with the latest optimization algorithms

In the last section, it can be observed that S-bBOA showed superior performance in comparison to the state-of-the-art approaches in terms of classification accuracy, average number of attributes selected as well as statistical measures (mean, best and worst) while S-bBOA demonstrated competitive performance in standard deviation fitness measure and average computational time. In the current section, in order to better validate the performance of the proposed approach, S-bBOA is compared with some latest high performing optimization algorithms namely DA, SCA, SSA and WOA. The classification accuracy produced using the selected features from S-bBOA, DA, SCA, SSA and WOA is presented in Table 17. It can be easily observed that the proposed approach (S-bBOA) outperformed all other optimizers on all the datasets utilized in this research work. This result substantiates the capability of the proposed S-bBOA to efficiently explore the solution search space and find the optimal feature subset with highest classification accuracy. From Table 18, the outstanding performance of the proposed S-bBOA approach in terms of selecting optimal feature subset can be observed. The proposed approach outperformed other algorithms on all the datasets except penglunEW which indicates that S-shaped function is able to significantly improve the performance of the native BOA in terms of selecting minimum number of attributes.

The statistical results when solving all the data sets are summarized in Tables 19-22. One can remark that S-bBOA outperforms DA, SCA, SSA and WOA in mean, best and worst performance measure while S-bBOA outperformed latest optimizers on most datasets (fifteen out of twenty-one datasets) in terms of standard deviation criteria. Table 23 outlines the average computational time of different optimization algorithms. It can be analyzed that WOA has minimum run time on nine datasets, whereas the proposed S-bBOA outperformed latest optimization algorithms on five datasets. On the other hand, DA, SCA and SSA performed better on four datasets each.

Overall, the results outlined in Tables 17-23 proved that S-bBOA has the ability to balance the exploration and exploitation during the optimization search process. It can be observed that the superior performance of S-bBOA is proved by small as well as large-sized datasets. There are four large-sized datasets; penglunEW, krvskpEW, clean2 and semeion are relatively large datasets

Table 17: Comparison between the proposed S-bBOA and latest optimization algorithms based on the classification accuracy

S. No	Dataset	DA	SCA	SSA	WOA	S-bBOA
1	Breastcancer	0.9626	0.9611	0.9600	0.9629	<b>0.9686</b>
2	BreastEW	0.9385	0.9396	0.9347	0.9425	<b>0.9709</b>
3	Clean1	0.8541	0.8585	0.8431	0.8499	<b>0.8832</b>
4	Clean2	0.9487	0.9510	0.9462	0.9517	<b>0.9614</b>
5	CongressEW	0.9318	0.9349	0.9321	0.9333	<b>0.9593</b>
6	Exactly	0.7481	0.7197	0.7091	0.7445	<b>0.9724</b>
7	Exactly2	0.7007	0.6977	0.6985	0.7011	<b>0.7596</b>
8	HeartEW	0.7773	0.7842	0.7595	0.7857	<b>0.8237</b>
9	IonosphereEW	0.8708	0.8826	0.8890	0.8848	<b>0.9070</b>
10	KrvskpEW	0.9269	0.8980	0.8929	0.9213	<b>0.9660</b>
11	Lymphography	0.7793	0.7880	0.7736	0.7658	<b>0.8676</b>
12	M-of-n	0.8293	0.8549	0.8361	0.8432	<b>0.9720</b>
13	PenglungEW	0.8268	0.7951	0.8331	0.8234	<b>0.8775</b>
14	Semeion	0.9721	0.9704	0.9649	0.9709	<b>0.9819</b>
15	SonarEW	0.8506	0.8506	0.8449	0.8532	<b>0.9362</b>
16	SpectEW	0.8000	0.7871	0.7826	0.7915	<b>0.8463</b>
17	Tic-tac-toe	0.7564	0.7546	0.7557	0.7610	<b>0.7983</b>
18	Vote	0.9227	0.9200	0.9196	0.9240	<b>0.9653</b>
19	WaveformEW	0.7154	0.7044	0.7091	0.7138	<b>0.7429</b>
20	WineEW	0.9551	0.9566	0.9506	0.9655	<b>0.9843</b>
21	Zoo	0.9359	0.9307	0.9476	0.9568	<b>0.9775</b>
<b>Average</b>		0.8573	0.8543	0.8516	0.8594	<b>0.9105</b>

Table 18: Comparison between the proposed S-bBOA and latest optimization algorithms based on average number of features

S. No	Dataset	DA	SCA	SSA	WOA	S-bBOA
1	Breastcancer	6.27	6.70	7.20	6.90	<b>5.60</b>
2	BreastEW	20.00	20.47	18.27	20.00	<b>16.80</b>
3	Clean1	109.67	110.20	94.87	121.93	<b>91.80</b>
4	Clean2	100.40	93.40	<b>90.40</b>	102.00	92.40
5	CongressEW	10.87	9.00	8.40	8.87	<b>6.40</b>
6	Exactly	10.53	10.47	12.80	11.20	<b>7.60</b>
7	Exactly2	8.67	9.00	6.27	9.47	<b>4.80</b>
8	HeartEW	9.60	8.47	7.47	9.80	<b>5.80</b>
9	IonosphereEW	18.00	19.07	19.67	21.67	<b>16.20</b>
10	KrvskpEW	28.60	30.80	36.00	27.60	<b>17.60</b>
11	Lymphography	12.53	10.87	12.20	14.20	<b>8.40</b>
12	M-of-n	12.13	10.67	12.33	10.67	<b>6.80</b>
13	PenglungEW	175.20	182.67	<b>162.33</b>	188.00	172.00
14	Semeion	193.00	194.40	161.80	181.40	<b>143.20</b>
15	SonarEW	40.60	37.13	34.13	40.13	<b>32.80</b>
16	SpectEW	14.67	12.60	11.33	14.13	<b>10.80</b>
17	Tic-tac-toe	7.20	7.47	8.07	7.87	<b>5.60</b>
18	Vote	8.87	9.60	8.53	10.33	<b>5.20</b>
19	WaveformEW	36.00	34.40	40.00	36.40	<b>25.00</b>
20	WineEW	9.53	9.40	9.07	9.93	<b>6.20</b>
21	Zoo	11.47	9.60	11.93	10.93	<b>5.20</b>

Table 19: Comparison between the proposed S-bBOA and latest optimization algorithms based on mean fitness measure

S. No	Dataset	DA	SCA	SSA	WOA	S-bBOA
1	Breastcancer	0.041	0.046	0.048	0.044	<b>0.040</b>
2	BreastEW	0.068	0.067	0.071	0.064	<b>0.042</b>
3	Clean1	0.151	0.147	0.160	0.156	<b>0.113</b>
4	Clean2	0.057	0.054	0.058	0.054	<b>0.051</b>
5	CongressEW	0.074	0.070	0.072	0.072	<b>0.045</b>
6	Exactly	0.257	0.286	0.298	0.262	<b>0.040</b>
7	Exactly2	0.303	0.306	0.303	0.303	<b>0.260</b>
8	HeartEW	0.228	0.220	0.244	0.220	<b>0.180</b>
9	IonosphereEW	0.133	0.122	0.116	0.120	<b>0.096</b>
10	KrvskpEW	0.080	0.110	0.116	0.086	<b>0.054</b>
11	Lymphography	0.225	0.216	0.231	0.240	<b>0.139</b>
12	M-of-n	0.178	0.152	0.172	0.163	<b>0.027</b>
13	PenglungEW	0.177	0.209	0.170	0.181	<b>0.118</b>
14	Semeion	0.035	0.037	0.040	0.036	<b>0.026</b>
15	SonarEW	0.155	0.154	0.159	0.152	<b>0.086</b>
16	SpectEW	0.205	0.217	0.220	0.213	<b>0.160</b>
17	Tic-tac-toe	0.249	0.251	0.251	0.245	<b>0.205</b>
18	Vote	0.082	0.085	0.085	0.082	<b>0.044</b>
19	WaveformEW	0.291	0.301	0.298	0.292	<b>0.265</b>
20	WineEW	0.052	0.050	0.056	0.042	<b>0.023</b>
21	Zoo	0.071	0.075	0.059	0.050	<b>0.034</b>

Table 20: Comparison between the proposed S-bBOA and latest optimization algorithms based on worst fitness measure

S. No	Dataset	DA	SCA	SSA	WOA	S-bBOA
1	Breastcancer	0.059	0.056	0.058	0.053	<b>0.041</b>
2	BreastEW	0.090	0.095	0.088	0.082	<b>0.049</b>
3	Clean1	0.178	0.214	0.200	0.188	<b>0.138</b>
4	Clean2	0.059	0.057	0.064	0.056	<b>0.048</b>
5	CongressEW	0.107	0.096	0.120	0.092	<b>0.058</b>
6	Exactly	0.319	0.375	0.335	0.306	<b>0.005</b>
7	Exactly2	0.330	0.337	0.363	0.328	<b>0.291</b>
8	HeartEW	0.284	0.277	0.299	0.262	<b>0.195</b>
9	IonosphereEW	0.157	0.155	0.157	0.159	<b>0.118</b>
10	KrvskpEW	0.097	0.176	0.121	0.110	<b>0.064</b>
11	Lymphography	0.303	0.261	0.299	0.314	<b>0.099</b>
12	M-of-n	0.210	0.236	0.212	0.240	<b>0.110</b>
13	PenglungEW	0.326	0.379	0.273	0.299	<b>0.169</b>
14	Semeion	0.040	0.047	0.045	0.042	<b>0.029</b>
15	SonarEW	0.217	0.198	0.214	0.223	<b>0.109</b>
16	SpectEW	0.252	0.271	0.262	0.292	<b>0.209</b>
17	Tic-tac-toe	0.293	0.307	0.293	0.277	<b>0.216</b>
18	Vote	0.124	0.138	0.118	0.121	<b>0.057</b>
19	WaveformEW	0.299	0.313	0.305	0.311	<b>0.265</b>
20	WineEW	0.086	0.077	0.076	0.076	<b>0.028</b>
21	Zoo	0.125	0.181	0.107	0.103	<b>0.048</b>

Table 21: Comparison between the proposed S-bBOA and latest optimization algorithms based on best fitness measure

S. No	Dataset	DA	SCA	SSA	WOA	S-bBOA
1	Breastcancer	0.031	0.038	0.038	0.037	<b>0.024</b>
2	BreastEW	0.051	0.052	0.059	0.045	<b>0.032</b>
3	Clean1	0.118	0.100	0.122	0.126	<b>0.088</b>
4	Clean2	0.050	0.052	0.054	0.050	<b>0.037</b>
5	CongressEW	0.035	0.045	0.041	0.057	<b>0.030</b>
6	Exactly	0.155	0.089	0.229	0.165	<b>0.005</b>
7	Exactly2	0.238	0.237	0.270	0.267	<b>0.225</b>
8	HeartEW	0.159	0.163	0.194	0.162	<b>0.138</b>
9	IonosphereEW	0.104	0.092	0.078	0.076	<b>0.060</b>
10	KrvskpEW	0.062	0.084	0.111	0.049	<b>0.036</b>
11	Lymphography	0.166	0.168	0.169	0.196	<b>0.073</b>
12	M-of-n	0.157	0.101	0.035	0.095	<b>0.005</b>
13	PenglungEW	0.035	0.062	0.112	0.088	<b>0.033</b>
14	Semeion	0.040	0.047	0.045	0.042	<b>0.029</b>
15	SonarEW	0.113	0.081	0.129	0.130	<b>0.072</b>
16	SpectEW	0.142	0.159	0.173	0.161	<b>0.122</b>
17	Tic-tac-toe	0.217	0.219	0.213	0.213	<b>0.195</b>
18	Vote	0.051	0.060	0.050	0.048	<b>0.016</b>
19	WaveformEW	0.278	0.291	0.291	0.273	<b>0.254</b>
20	WineEW	0.031	0.028	0.016	0.008	<b>0.005</b>
21	Zoo	0.007	0.009	0.026	0.009	<b>0.002</b>

Table 22: Comparison between the proposed S-bBOA and latest optimization algorithms based on standard deviation measure

S. No	Dataset	DA	SCA	SSA	WOA	S-bBOA
1	Breastcancer	0.011	0.013	0.008	0.012	<b>0.006</b>
2	BreastEW	0.01	0.005	0.005	0.005	<b>0.003</b>
3	Clean1	0.021	0.016	0.02	0.013	<b>0.01</b>
4	Clean2	0.038	0.066	0.025	0.04	<b>0.01</b>
5	CongressEW	0.022	0.023	0.025	<b>0.016</b>	0.02
6	Exactly	0.028	0.035	0.028	0.032	<b>0.011</b>
7	Exactly2	<b>0.015</b>	0.019	0.021	0.02	0.059
8	HeartEW	0.014	0.037	<b>0.005</b>	0.025	0.008
9	IonosphereEW	0.041	0.032	0.042	0.031	<b>0.01</b>
10	KrvskpEW	0.018	0.035	0.049	0.04	<b>0.003</b>
11	Lymphography	0.023	0.03	0.021	0.024	<b>0.018</b>
12	M-of-n	0.03	0.031	<b>0.025</b>	0.037	0.035
13	PenglungEW	0.023	0.025	0.021	0.02	<b>0.007</b>
14	Semeion	0.019	0.024	0.02	0.023	<b>0.01</b>
15	SonarEW	0.009	0.009	0.006	0.014	<b>0.005</b>
16	SpectEW	0.013	0.014	0.015	0.019	<b>0.01</b>
17	Tic-tac-toe	0.037	0.047	0.028	<b>0.023</b>	0.044
18	Vote	0.018	0.029	0.022	0.02	<b>0.016</b>
19	WaveformEW	0.003	0.002	0.004	0.003	<b>0.001</b>
20	WineEW	0.077	0.093	<b>0.052</b>	0.062	0.056
21	Zoo	<b>0.003</b>	0.006	0.004	0.006	0.004

Table 23: Comparison between the proposed S-bBOA and latest optimization algorithms based on the average running time

S. No	Dataset	DA	SCA	SSA	WOA	S-bBOA
1	Breastcancer	2.48	2.36	2.32	<b>2.31</b>	2.36
2	BreastEW	2.82	2.41	2.36	<b>2.35</b>	2.41
3	Clean1	4.05	3.61	<b>3.5</b>	3.54	3.75
4	Clean2	<b>171.84</b>	223.7	173.07	182.94	236.46
5	CongressEW	3.03	2.59	<b>2.61</b>	<b>2.59</b>	2.63
6	Exactly	<b>4.06</b>	4.63	5.18	4.65	4.87
7	Exactly2	4.6	4.88	4.2	4.22	<b>4.11</b>
8	HeartEW	2.78	2.52	2.62	2.47	<b>2.41</b>
9	IonosphereEW	2.96	<b>2.6</b>	<b>2.54</b>	2.57	2.61
10	KrvskpEW	13.84	<b>15.89</b>	13.89	<b>13.03</b>	15.16
11	Lymphography	2.82	<b>2.38</b>	2.87	2.91	2.78
12	M-of-n	4.04	<b>3.74</b>	4.26	4.14	4.28
13	PenglungEW	4.13	2.55	<b>2.5</b>	<b>2.5</b>	2.54
14	Semeion	19.92	24.06	21.82	<b>19.21</b>	23.34
15	SonarEW	2.92	2.72	2.75	2.59	<b>2.54</b>
16	SpectEW	2.8	<b>2.38</b>	2.4	<b>2.38</b>	<b>2.38</b>
17	Tic-tac-toe	<b>3.91</b>	4.38	4.25	4.1	4.54
18	Vote	2.82	2.6	2.53	<b>2.47</b>	2.53
19	WaveformEW	<b>35.56</b>	43.72	84.14	86.64	89.42
20	WineEW	2.68	<b>2.43</b>	<b>2.43</b>	2.47	<b>2.43</b>
21	Zoo	2.66	2.3	2.21	<b>2.19</b>	2.24

Table 24:  $p$ -values of the Wilcoxon test of the proposed S-bBOA vs. other algorithms ( $p \geq 0.05$  are underlined)

S. No	Dataset	ALO	BSO	GA	GWO	PSO	DA	SCA	SSA	WOA	BOA	V-bBOA
1	Breast Cancer	1.09E-02	5.06E-03	5.03E-03	6.91E-03	1.09E-02	<u>1.14E-01</u>	6.87E-03	5.06E-03	1.52E-02	7.63E-03	6.23E-04
2	BreastEW	6.48E-04	6.55E-04	6.32E-04	6.53E-04	6.43E-04	6.55E-04	6.52E-04	6.55E-04	9.82E-04	6.45E-04	7.63E-03
3	Clean1	6.55E-04	9.85E-04	2.15E-03	8.01E-04	2.15E-03	8.03E-04	1.79E-03	6.53E-04	7.99E-04	4.51E-03	<u>7.21E-01</u>
4	Clean2	1.03E-02	6.23E-04	6.23E-04	6.23E-04	6.23E-04	2.07E-03	6.23E-04	6.23E-04	1.03E-02	6.23E-04	6.23E-04
5	CongressEW	9.85E-04	6.55E-04	1.19E-03	1.21E-03	6.55E-04	4.51E-03	2.16E-03	1.47E-03	6.55E-04	4.51E-03	<u>3.44E-01</u>
6	Exactly	6.52E-04	6.55E-04	6.42E-04	6.48E-04	6.55E-04	6.53E-04	6.55E-04	6.52E-04	6.55E-04	6.55E-04	<u>8.54E-01</u>
7	Exactly2	9.87E-04	6.55E-04	6.47E-04	6.55E-04	8.05E-04	1.21E-03	8.05E-04	8.05E-04	6.55E-04	8.98E-03	8.98E-03
8	HeartEW	9.87E-04	6.53E-04	6.47E-04	9.79E-04	6.53E-04	8.05E-04	6.41E-03	6.53E-04	1.89E-03	7.59E-03	8.98E-03
9	IonosphereEW	3.09E-02	2.31E-02	<u>8.20E-01</u>	<u>7.83E-02</u>	<u>3.34E-01</u>	<u>6.09E-02</u>	<u>3.07E-01</u>	<u>5.32E-01</u>	<u>3.63E-01</u>	<u>4.60E-01</u>	<u>5.35E-02</u>
10	KrvskpEW	4.31E-02	4.31E-02	<u>7.96E-02</u>	4.31E-02	<u>7.96E-02</u>	4.31E-02	4.31E-02	4.31E-02	<u>7.96E-02</u>	4.31E-02	4.31E-02
11	Lymphography	6.55E-04	6.55E-04	6.47E-04	6.55E-04	6.55E-04	6.55E-04	6.52E-04	6.55E-04	6.55E-04	6.55E-04	6.23E-04
12	M-of-n	6.55E-04	6.55E-04	6.53E-04	6.53E-04	6.55E-04	6.55E-04	6.55E-04	6.53E-04	6.55E-04	6.55E-04	6.55E-04
13	PenglungEW	1.71E-02	3.14E-03	<u>1.40E-01</u>	<u>8.83E-02</u>	3.09E-02	<u>7.83E-02</u>	3.56E-02	2.68E-02	1.70E-02	<u>4.95E-01</u>	<u>3.05E-01</u>
14	semeion	<u>1.38E-01</u>	4.31E-02	<u>7.96E-02</u>	4.31E-02							
15	SonarEW	6.55E-04	6.55E-04	6.53E-04	6.55E-04	1.79E-03	6.55E-04	6.55E-04	6.53E-04	6.55E-04	6.55E-04	6.55E-04
16	SpectEW	1.25E-02	6.55E-04	4.48E-03	6.55E-04	6.55E-04	5.37E-03	1.79E-03	6.55E-04	1.79E-03	1.46E-02	<u>8.64E-01</u>
17	Tic-tac-toe	6.50E-04	6.55E-04	6.53E-04	6.55E-04	8.03E-04	6.55E-04	6.55E-04	6.53E-04	6.53E-04	8.05E-04	1.46E-02
18	Vote	9.85E-04	6.53E-04	6.47E-04	9.87E-04	6.53E-04	6.55E-04	6.55E-04	6.55E-04	1.21E-03	6.53E-04	<u>2.31E-01</u>
19	WaveformEW	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	<u>2.23E-01</u>	<u>2.25E-01</u>
20	WineEW	6.48E-04	6.53E-04	6.50E-04	3.77E-03	8.03E-04	6.52E-04	8.01E-04	6.48E-04	7.59E-03	1.19E-03	4.31E-02
21	Zoo	3.56E-02	3.14E-03	4.67E-02	<u>6.09E-01</u>	<u>3.07E-01</u>	1.40E-01	<u>6.91E-02</u>	<u>3.63E-01</u>	<u>5.94E-01</u>	<u>9.55E-01</u>	<u>2.36E-01</u>

and the classification accuracy, average selection size and the fitness measures (mean, best, worst and standard deviation) of the proposed approach is evidently better than all the other algorithms.

Moreover the *p*-values of the Wilcoxon's rank sum test obtained at 5% significance level are reported in Table 24. The *p*-values less than 0.05 indicate that the null hypothesis is rejected, i.e., there is significant difference at a level of 5%. The *p*-values in Table 24 confirm that the results of the proposed approach S-bBOA are significantly different from the results of the state-of-the-art as well as latest algorithms on majority of the datasets. Furthermore, in order to provide a fair comparison, the algorithms are compared on basis of their computational complexity. The computational complexity of BOA is of  $O(i(v \times s + C \times s))$  where  $i$  represents the number of iterations,  $v$  shows the number of variables,  $C$  is the cost of the objective function and  $s$  indicates the number of solutions. This computation complexity is equal to that of DA, PSO, SCA and WOA. However, the computational complexity of ALO, BSO, GA, GWO and SSA is of  $O(i \times (v \times s + C \times s + s \times \log(s)))$  in the best case and  $O(i \times (v \times s + C \times s + s^2))$  in the worst case. This means that the computational complexity of ALO, BSO, GA, GWO and SSA is worse than those of BOA, DA, PSO, SCA and WOA due to the need to sort the solutions in each iteration.

Taken together, the simulation results presented in this section proved that the BOA is an effective stochastic wrapper-based feature selection method which has the ability to solve a broader range of classification problems. Basically, the optimization is different from the continuous search space where the search agents are allowed to move around. In order to change the variable from 0 to 1 and vice versa, a very high exploration is required. Moreover, the algorithm must possess the ability to avoid local optima, therefore it should alter the variables of a binary problem less frequently with respect to the number of iterations. The adaptive mechanism in the BOA algorithm accelerates the convergence speed with respect to the number of iterations. It also plays a vital role in the balance of exploration and exploitation in order to avoid a large number of local solutions in feature selection problems, and then to find an accurate estimation of the best solution. This is the main reason for the superior performance of BOA in comparison to the other algorithms used in this study.

We can remark that the bBOA provides maximum accuracy and minimum

number of features; however, it is worthwhile to mention here is that bBOA requires more computational time in comparison to other algorithms used in this study, except ALO. Another limitation of the proposed stochastic wrapper-based feature selection approach is the non-exact repeatability of the optimization results. It has been observed that the subset of features selected by the algorithm might differ for different applications which may be confusing for the user to decide which subset to consider. The proposed approach uses KNN classifier which is a simple one and the running time may increase when switched to another classifier such as support vector machine or random forest. Thus, a different classifier should be carefully handled, especially in the case of real-world applications.

## 6. Conclusion

In this research work, binary variants of the butterfly optimization algorithm (BOA) are proposed and utilized to solve feature selection problem in wrapper mode. The continuous version of butterfly optimization algorithm is transformed into the binary form using either V-shaped or S-shaped transfer functions. The proposed approaches are employed and used for feature selection in the domain of machine learning to assess different searching capabilities of the algorithms. The feature selection problem is formulated as a multi-objective optimization task with a fitness function reflecting the classification performance and the reduction in the number of features. Twenty-one well-known datasets from the UCI repository were used to assess the performance of the proposed approaches. The proposed binary butterfly optimization algorithms are employed in the feature selection field for assessment and the simulation results are compared against various state-of-the-art feature selection approaches namely Genetic Algorithm (GA), Particle Swarm Optimizer (PSO), Ant Lion Optimizer (ALO), Brain Storm Optimizer (BSO) and Grey Wolf Optimizer (GWO). Furthermore, the proposed binary approaches are further compared with four latest and high performing optimization algorithms namely Dragonfly Algorithm (DA), Sine-Cosine Algorithm (SCA), Salp Swarm Algorithm (SSA) and Whale Optimization Algorithm (WOA). The evaluation is performed using a set of evaluation criteria to assess different aspects of performance. Experimental

results showed that our approaches are able to produce better results than the other approaches on most of the datasets. Moreover, the results show that the BOA with S-shaped function can significantly enhance the performance BOA in terms of classification accuracy and number of selected features. The summarized simulation results depict that the proposed S-bBOA algorithm has the ability to effectively search the feature set space and converge to (near) optimal solution better than the other algorithms.

For future studies, binary BOA can be applied to various other public datasets, real-world problems and it can also be employed with more classifiers like artificial neural networks, k-nearest neighbors and support vector machines to extend the current approach. Furthermore, it would be interesting to hybridize BOA algorithm with another population-based metaheuristic algorithm like GWO or ALO. It would be interesting to investigate the performance of BOA when applied to many higher-dimensional datasets. Additionally, a study on the effect of high dimensionality and small-sized samples versus large-sized samples with respect to the proposed method is an interesting work for the future. Moreover, enhanced initialization method can be utilized that starts the optimization with solution pool closer to optimal which could improve the computational time.

## 7. Acknowledgment

We acknowledge the valuable comments of anonymous reviewers. We would also like to thank the Editors for their generous comments and support during the review process.

## References

- Ahmad, S., Mafarja, M., Faris, H., & Aljarrah, I. (2018). Feature selection using salp swarm algorithm with chaos, .
- Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46, 175–185.

- Arora, S., & Singh, S. (2015). Butterfly algorithm with levy flights for global optimization. In *International Conference on Signal Processing, Computing and Control* (pp. 220–224). Solan, Himachal Pradesh, India: IEEE.
- Arora, S., & Singh, S. (2016). An improved butterfly optimization algorithm for global optimization. *Advanced Science, Engineering and Medicine*, 8, 711–717.
- Arora, S., & Singh, S. (2017a). An effective hybrid butterfly optimization algorithm with artificial bee colony for numerical optimization. *International Journal of Interactive Multimedia and Artificial Intelligence*, 4, 14–21.
- Arora, S., & Singh, S. (2017b). A hybrid optimisation algorithm based on butterfly optimisation algorithm and differential evolution. *International Journal of Swarm Intelligence*, 3, 152–169.
- Arora, S., & Singh, S. (2017c). An improved butterfly optimization algorithm with chaos. *Journal of Intelligent & Fuzzy Systems*, 32, 1079–1088.
- Arora, S., & Singh, S. (2017d). Node localization in wireless sensor networks using butterfly optimization algorithm. *Arabian Journal for Science and Engineering*, 42, 3325–3335.
- Arora, S., & Singh, S. (2018). Butterfly optimization algorithm: a novel approach for global optimization. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, . doi:10.1007/s00500-018-3102-4.
- Arora, S., Singh, S., & Yetilmezsoy, K. (2018). A modified butterfly optimization algorithm for mechanical design optimization problems. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 40, 21.
- Bello, R., Gomez, Y., Nowe, A., & Garcia, M. M. (2007). Two-step particle swarm optimization to solve the feature selection problem. In *Intelligent Systems Design and Applications, 2007. ISDA 2007. Seventh International Conference on* (pp. 691–696). IEEE.
- Bennasar, M., Hicks, Y., & Setchi, R. (2015). Feature selection using joint mutual information maximisation. *Expert Systems with Applications*, 42, 8520–8532.

- Blair, R. B., & Launer, A. E. (1997). Butterfly diversity and human land use: Species assemblages along an urban gradient. *Biological Conservation*, 80, 113–125.
- Cui, Z., Sun, B., Wang, G., Xue, Y., & Chen, J. (2017). A novel oriented cuckoo search algorithm to improve dv-hop performance for cyber-physical systems. *Journal of Parallel and Distributed Computing*, 103, 42–52.
- Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1, 3–18.
- Eberhart, R., & Kennedy, J. (1995). Proceedings of ieee international conference on neural networks, .
- Emary, E., & Zawbaa, H. M. (2018). Feature selection via lèvy antlion optimization. *Pattern Analysis and Applications*, (pp. 1–20).
- Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016a). Binary ant lion approaches for feature selection. *Neurocomputing*, 213, 54–65.
- Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016b). Binary grey wolf optimization approaches for feature selection. *Neurocomputing*, 172, 371–381.
- Faris, H., Mafarja, M. M., Heidari, A. A., Aljarah, I., AlaM, A.-Z., Mirjalili, S., & Fujita, H. (2018). An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowledge-Based Systems*, .
- Feng, Y., & Wang, G.-G. (2018). Binary moth search algorithm for discounted 0-1 knapsack problem. *IEEE Access*, 6, 10708–10719.
- Feng, Y., Wang, G.-G., Deb, S., Lu, M., & Zhao, X.-J. (2017). Solving 0-1 knapsack problem by a novel binary monarch butterfly optimization. *Neural computing and applications*, 28, 1619–1634.
- Feng, Y., Yang, J., Wu, C., Lu, M., & Zhao, X.-J. (2018). Solving 0-1 knapsack problems by chaotic monarch butterfly optimization algorithm with gaussian mutation. *Memetic Computing*, 10, 135–150.

- Gandomi, A. H., & Alavi, A. H. (2012). Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 17, 4831–4845.
- Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.
- Hastie, T., Tibshirani, R., & Friedman, J. (2001). The elements of statistical learning new york. NY: Springer, (pp. 115–163).
- Hedar, A.-R., Wang, J., & Fukushima, M. (2008). Tabu search for attribute reduction in rough set theory. *Soft Computing*, 12, 909–918.
- Holland, J., & Goldberg, D. (1989). Genetic algorithms in search, optimization and machine learning. Massachusetts: Addison-Wesley, .
- Jensen, R., & Shen, Q. (2004). Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. *IEEE Transactions on knowledge and data engineering*, 16, 1457–1471.
- Kabir, M. M., Shahjahan, M., & Murase, K. (2011). A new local search based hybrid genetic algorithm for feature selection. *Neurocomputing*, 74, 2914–2928.
- Kashef, S., & Nezamabadi-pour, H. (2015). An advanced aco algorithm for feature subset selection. *Neurocomputing*, 147, 271–279.
- Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on* (pp. 4104–4108). IEEE volume 5.
- Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering*, 17, 491–502.
- Mafarja, M., & Abdullah, S. (2013). Record-to-record travel algorithm for attribute reduction in rough set theory. *J Theor Appl Inf Technol*, 49, 507–513.

- Mafarja, M., & Abdullah, S. (2015). A fuzzy record-to-record travel algorithm for solving rough set attribute reduction. *International Journal of Systems Science*, 46, 503–512.
- Mafarja, M., & Mirjalili, S. (2018). Whale optimization approaches for wrapper feature selection. *Applied Soft Computing*, 62, 441–453.
- Mafarja, M. M., Eleyan, D., Jaber, I., Hammouri, A., & Mirjalili, S. (2017). Binary dragonfly algorithm for feature selection. In *New Trends in Computing Sciences (ICTCS), 2017 International Conference on* (pp. 12–17). IEEE.
- Mafarja, M. M., & Mirjalili, S. (2017). Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing*, 260, 302–312.
- Mirjalili, S. (2016a). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, 27, 1053–1073.
- Mirjalili, S. (2016b). Sca: a sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, 96, 120–133.
- Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 114, 163–191.
- Mirjalili, S., & Hashim, S. Z. M. (2012). Bmoa: binary magnetic optimization algorithm. *International Journal of Machine Learning and Computing*, 2, 204.
- Mirjalili, S., & Lewis, A. (2013). S-shaped versus v-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, 9, 1–14.
- Mirjalili, S., Mirjalili, S. M., & Yang, X.-S. (2014). Binary bat algorithm. *Neural Computing and Applications*, 25, 663–681.
- Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy.

*IEEE Transactions on pattern analysis and machine intelligence*, 27, 1226–1238.

- Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2010). Bgsa: binary gravitational search algorithm. *Natural Computing*, 9, 727–745.
- Rizk-Allah, R. M., El-Sehiemy, R. A., Deb, S., & Wang, G.-G. (2017). A novel fruit fly framework for multi-objective shape design of tubular linear synchronous motor. *The Journal of Supercomputing*, 73, 1235–1256.
- Rizk-Allah, R. M., El-Sehiemy, R. A., & Wang, G.-G. (2018). A novel parallel hurricane optimization algorithm for secure emission/economic load dispatch solution. *Applied Soft Computing*, 63, 206–222.
- Sayed, G. I., Khoriba, G., & Haggag, M. H. (2018). A novel chaotic salp swarm algorithm for global optimization and feature selection. *Applied Intelligence*, (pp. 1–20).
- Shi, Y. (2011). Brain storm optimization algorithm. In *International Conference in Swarm Intelligence* (pp. 303–309). Springer.
- Stevens, S. S. (1975). *Psychophysics*. Transaction Publishers, United Kingdom.
- Wang, G.-G. (2016). Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Computing*, (pp. 1–14).
- Wang, G.-G., Cai, X., Cui, Z., Min, G., & Chen, J. (2017). High performance computing for cyber physical social systems by using evolutionary multi-objective optimization algorithm. *IEEE Transactions on Emerging Topics in Computing*,
- Wang, G.-G., Chu, H. E., & Mirjalili, S. (2016a). Three-dimensional path planning for uav using an improved bat algorithm. *Aerospace Science and Technology*, 49, 231–238.
- Wang, G.-G., Deb, S., & Cui, Z. (2015). Monarch butterfly optimization. *Neural Computing and Applications*, (pp. 1–20).
- Wang, G.-G., Deb, S., Gandomi, A. H., Zhang, Z., & Alavi, A. H. (2016b). Chaotic cuckoo search. *Soft Computing*, 20, 3349–3362.

- Wang, G.-G., Deb, S., Gao, X.-Z., & Coelho, L. D. S. (2016c). A new meta-heuristic optimisation algorithm motivated by elephant herding behaviour. *International Journal of Bio-Inspired Computation*, 8, 394–409.
- Wang, G.-G., Deb, S., Zhao, X., & Cui, Z. (2016d). A new monarch butterfly optimization with an improved crossover operator. *Operational Research*, (pp. 1–25).
- Wang, G.-G., Gandomi, A. H., & Alavi, A. H. (2014a). Stud krill herd algorithm. *Neurocomputing*, 128, 363–370.
- Wang, G.-G., Guo, L., Gandomi, A. H., Hao, G.-S., & Wang, H. (2014b). Chaotic krill herd algorithm. *Information Sciences*, 274, 17–34.
- Wang, J., Li, T., & Ren, R. (2010). A real time idss based on artificial bee colony-support vector machine algorithm. In *Advanced computational intelligence (IWACI), 2010 third international workshop on* (pp. 91–96). IEEE.
- Wu, G., Pedrycz, W., Li, H., Ma, M., & Liu, J. (2016). Coordinated planning of heterogeneous earth observation resources. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46, 109–125.
- Yang, X.-S. (2010a). Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, 2, 78–84.
- Yang, X.-S. (2010b). *Nature-inspired metaheuristic algorithms*. Luniver press.
- Yang, X.-S., & Deb, S. (2009). Cuckoo search via lévy flights. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on* (pp. 210–214). IEEE.
- Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Icml* (pp. 412–420). volume 97.
- Yi, J.-H., Wang, J., & Wang, G.-G. (2016). Improved probabilistic neural networks with self-adaptive strategies for transformer fault diagnosis problem. *Advances in Mechanical Engineering*, 8, 1687814015624832.
- Zawbaa, H. M., & Emary, E. (2018). Applications of flower pollination algorithm in feature selection and knapsack problems. In *Nature-Inspired Algorithms and Applied Optimization* (pp. 217–243). Springer.

Zawbaa, H. M., Emary, E., & Grosan, C. (2016). Feature selection via chaotic antlion optimization. *PloS one*, 11, e0150652.

Zawbaa, H. M., Emary, E., Grosan, C., & Snasel, V. (2018). Large-dimensionality small-instance set feature selection: A hybrid bio-inspired heuristic approach. *Swarm and Evolutionary Computation*, .

Zhang, J. W., & Wang, G. G. (2012). Image matching using a bat algorithm with mutation. In *Applied Mechanics and Materials* (pp. 88–93). Trans Tech Publ volume 203.

Zorarpacı, E., & Özal, S. A. (2016). A hybrid approach of differential evolution and artificial bee colony for feature selection. *Expert Systems with Applications*, 62, 91–103.