



An improved Dragonfly Algorithm for feature selection

Abdelaziz I. Hammouri^{a,*}, Majdi Mafarja^b, Mohammed Azmi Al-Betar^c,
Mohammed A. Awadallah^d, Iyad Abu-Doush^{e,f}

^a Department of Computer Information Systems, Al-Balqa Applied University, 19117, Al-Salt, Jordan

^b Department of Computer Science, Birzeit University, Birzeit, Palestine

^c Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, Al-Huson, Irbid, Jordan

^d Department of Computer Science, Al-Aqsa University, P.O. Box 4051, Gaza, Palestine

^e Computer Science Department, Yarmouk University, Irbid, Jordan

^f Computing Department, American University of Kuwait, Salmiya, Kuwait

ARTICLE INFO

Article history:

Received 14 December 2019

Received in revised form 6 June 2020

Accepted 9 June 2020

Available online 19 June 2020

Keywords:

Feature selection

Binary Dragonfly Algorithm

Selected features

Classification accuracy

V-shaped transfer function

Optimization

ABSTRACT

Dragonfly Algorithm (DA) is a recent swarm-based optimization method that imitates the hunting and migration mechanisms of idealized dragonflies. Recently, a binary DA (BDA) has been proposed. During the algorithm iterative process, the BDA updates its five main coefficients using random values. This updating mechanism can be improved to utilize the survival-of-the-fittest principle by adopting different functions such as linear, quadratic, and sinusoidal. In this paper, a novel BDA is proposed. The algorithm uses different strategies to update the values of its five main coefficients to tackle Feature Selection (FS) problems. Three versions of BDA have been proposed and compared against the original DA. The proposed algorithms are Linear-BDA, Quadratic-BDA, and Sinusoidal-BDA. The algorithms are evaluated using 18 well-known datasets. Thereafter, they are compared in terms of classification accuracy, the number of selected features, and fitness value. The results show that Sinusoidal-BDA outperforms other proposed methods in almost all datasets. Furthermore, Sinusoidal-BDA exceeds three swarm-based methods in all the datasets in terms of classification accuracy and it excels in most datasets when compared in terms of the fitness function value. In a nutshell, the proposed Sinusoidal-BDA outperforms the comparable feature selection algorithms and the proposed updating mechanism has a high impact on the algorithm performance when tackling FS problems.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

The use of computer technologies and information systems grows rapidly and a huge amount of data is generated daily using different devices. Data mining (DM) [1] is a confluence between multiple disciplines such as databases, machine learning, pattern recognition, algorithms, and statistics, that aims to turn the vast amount of data into useful information. In order to use data mining effectively, data pre-processing is applied to clean and prepare the data for the next phases of machine learning [2]. FS is a commonly used method in data pre-processing for data mining [3]. FS aims to select the smallest subset of relevant features from a dataset that best captures data properties for adequate classification. This can be done by eliminating repeated, unnecessary, or noisy features. FS can speed up DM techniques, as well as, improving their performance.

FS is an important phase of data mining. It is applied as the first steps in solving many problems such as data classification [4–7], data clustering [8–10], image processing [11,12], music composition [13], and text categorization [14,15]. In general, FS methods can be categorized based on two major aspects: the evaluation criterion and the search strategy. From the evaluation criterion perspective, there are two main types of FS methods filter technique and wrapper technique. The difference between these two techniques is the use of the learning algorithm (e.g., classifiers) when evaluating the generated feature subsets. Filter methods rely on the correlations between the conditional features and the class without using a specific learning algorithm (e.g., Chi-Square [16], Information Gain [17], Gain Ratio [18], Support Vector Machine [19], ReliefF [20,21], and hybrid ReliefF [22,23]). A learning algorithm is applied in the evaluation process of wrapper methods [24]. As a result, wrapper methods generate more accurate results than filter methods. On the other hand, the wrapper approach may require more computational resources than the filter approach. Furthermore, the performance of the wrapper method depends on the employed learning algorithm [25].

* Corresponding author.

E-mail addresses: Aziz@bau.edu.jo (A.I. Hammouri), mmafarja@birzeit.edu (M. Mafarja), mohbetar@bau.edu.jo (M.A. Al-Betar), ma.awadallah@alaqsa.edu.ps (M.A. Awadallah), idoush@auk.edu.kw (I. Abu-Doush).

The FS process is considered as an NP-hard searching problem [26,27]. Thus, when dealing with high dimensional data, searching for the optimal (minimal) feature subset is not an easy task. For a dataset with N features, 2^N subsets need to be tested to determine the best feature subset that can be used to differentiate data records [3]. Hence, for the high dimensional data, exhaustively generating all possible subsets becomes impractical and computationally expensive. Sequential forward, sequential backward, random, and heuristic search are all examples of search strategies that can be used to improve FS method efficiency. Recently, metaheuristics algorithms are successfully applied with many FS methods [28].

Metaheuristics algorithms are successful in solving various optimization problems when compared to exact methods [3,28,29]. Swarm Intelligence (SI) is metaheuristic algorithms that were inspired by some natural phenomena like the group cooperating behavior of the ants, bees, flocks of birds, dragonflies, etc. [30]. Particle Swarm Optimization (PSO) [31], Ant Colony Optimization (ACO) [32,33], Artificial Bees Colony (ABC) [34], Gravitational Search Algorithm [35], Grey Wolf Optimizer (GWO) [36–38], Harmony Search Algorithm [39], Whale Optimization Algorithm (WOA) [40,41], Salp Swarm Algorithm (SSA) [42–44] are examples of SI algorithms that applied to solve FS optimization problems.

Recently, new SI algorithms have been applied to the FS problem. Antlion Optimizer (ALO) [45] is applied as a wrapper method for FS method [46,47]. The GWO is another SI algorithm that has been successfully applied to FS problems [48]. Moreover, a WOA approach is proposed in [49], where the Simulated Annealing (SA) was used to enhance the neighborhood searchability of the WOA in a hybrid model. Moreover, other FS approaches are proposed in [50,51] and [52] based on the WOA algorithm. Most of the previous approaches discussed the importance of having a good balance between exploration and exploitation. However, according to the No-Free-Lunch theorem [53], the door is still open for more improvements due to the stochastic nature of the metaheuristics algorithms.

Dragonfly Algorithm (DA) Mirjalili [54] is a recent SI algorithm that mimics the five primitive principles of the swarming behavior of dragonflies. The dragonflies rely on separation to avoid collisions between individuals in the swarm; alignment to maintain the velocity of all individuals in the swarm; cohesion to relate dragonflies to a neighborhood; attraction to move dragonflies towards the food source; distraction to move dragonflies far from the enemies. By maintaining these five principles dragonflies can reach the food source and thus survive. These five principles require five parameters to be controlled, that are separation S_i , alignment A_i , cohesion C_i , attraction towards the food source F_i , and distraction from the enemy E_i .

The DA is applied successfully to solve several optimization problems such as economic dispatch [55], hybrid energy distributed power system [56], power flow management [57], image segmentation [58,59], stress distribution [60], synthesis of concentric circular antenna arrays [61], structural optimization of frame structures [62], structural design optimization of vehicle components [63], filter design problem [64], infant cry classification [65], internet of vehicles [66], traveling salesman problem [67], wireless node localization in computer networks [68], 0–1 knapsack problems [69], and machine learning [70]. A binary version of DA (called BDA) was proposed in Mirjalili [54]. In BDA, a transfer function (TF) was used to map the continuous search space into binary. BDA was initially applied to the FS problem in [71], and the method produces high-quality results. Recently, a novel FS approach the employs an enhanced BDA was proposed in [70]. In DA, the balance between exploration and exploitation depends mainly on some operators that gradually

decrease or increase in the course of iterations. These operators depend on the updating mechanisms of the main five coefficients of the algorithm during the search process. However, in the original DA, updating the main five coefficients values is done based on a random function which directly affects the algorithm convergence.

FS is a very important problem in data mining. Recently the BDA is proposed and it achieves superior results that outperform all comparative methods [70]. In this paper, novel modified binary Dragonfly Algorithms (DA) with different strategies to update the values of its five main coefficients (i.e., s , a , c , f , and e) are proposed to tackle FS problem. These coefficients have a major role to find the right balance between exploration and exploitation during the search process. Three versions of BDA have been proposed: (i) Linear based Binary Dragonfly (LBDA), (ii) Quadratic based Binary Dragonfly (QBDA), and (iii) Sinusoidal based Binary Dragonfly (SBDA). The updating mechanism of these versions of BDA is based on linear, quadratic, and sinusoidal functions. The proposed algorithms are compared against Random based Binary Dragonfly (RBDA) which is the original BDA and with four swarm based algorithms.

These versions are evaluated using 18 well-known datasets selected from UCI repository. Initially, the effect of the angle control parameter called β in SBDA is studied. Thereafter, the four proposed versions are compared in terms of three evaluation measures average classification accuracy, the number of the selected features, and fitness value. The SBDA is able to excel the other proposed methods in almost all datasets. Furthermore, SBDA is able to outperform other four swarm-based methods in all datasets in terms of average classification accuracy, 8 out of 18 datasets in terms of average selected features, and 16 out of 18 in terms of fitness function values. In comparison with nine state-of-the-art methods, the proposed SBDA version is able to yield 12 out of 18 best results in terms of average classification accuracy.

DA is a recent metaheuristic algorithm that shows excellent performance in tackling different optimization problems (e.g., continuous, discrete, single-objective, and multi-objective) when compared to several state-of-the-art metaheuristic and evolutionary algorithms. However, the algorithm has been used with FS in a few previous research works. This motivates proposing a new approach when applying this algorithm. The main objective of this paper is to propose three improved versions of Binary Dragonfly Algorithm (BDA) to updating its coefficients based three different migration models. These are (i) Linear based Binary Dragonfly (LBDA), and (ii) Quadratic based Binary Dragonfly (QBDA), and (iii) Sinusoidal based Binary Dragonfly (SBDA). To achieve such objective, the following sub-objectives are considered:

- Investigating the influence of the traditional updating mechanisms of the BDA coefficients on exploration and exploitation process.
- Three advanced updating mechanisms are proposed to update the coefficients of the BDA and thus to overcome the drawbacks of the traditional mechanisms.
- A set of 18 well-known datasets, that are widely used by the literature, were adopted to assess the performance of the proposed BDA versions along with the original version.
- A comparative evaluation is conducted against 11 state-of-the-methods using the same datasets.

Interestingly, the proposed SBDA is able to obtain four new results and five best results as achieved by other comparative in terms of accuracy. Furthermore, the proposed LBDA is able to produce three new results and five best results as produced by others. The proposed QBDA is able to produce two new results

and three best results as produced by others. The obtained results revealed that the ability of the proposed updating mechanisms are very efficient in improving the search capabilities of the BDA.

The remaining sections of this paper are organized as follows: a preliminary background about the DA algorithms is presented in Section 2. Section 3 presents the details of the binary version of the DA. The proposed approach is presented in Section 4. The experimental setup, the obtained results, the comparisons against other algorithms, and the discussions are reported in Section 5. Finally, Section 6 presents the conclusions and draws some future directions.

2. The Dragonfly Algorithm

Dragonfly Algorithm (DA) Mirjalili [54] is a recently proposed population-based metaheuristic algorithm. The DA mainly inspires the behavior of hunting (called static swarm (feeding)) and migration mechanisms of idealized dragonflies. In nature, the dragonflies fly in small groups seeking food sources. This process is called hunting mechanism. Larger groups of dragonflies fly with each other in one direction, so that the swarm migrates in a process called migration mechanism. Both mechanisms of hunting and feeding are illustrated in Fig. 1.

The dragonflies swarming behavior is characterized by five operators [72]:

- *Separation* is the mechanism that ensures keeping the search agents away from each other in the neighborhood. The mathematical modeling of the separation behavior is shown in Eq. (1).

$$S_i = - \sum_{j=1}^N X - X_i \quad (1)$$

- *Alignment* indicates how the velocity of a specific search agent is matched with the velocity of other search agents in the neighborhood. The mathematical modeling of the alignment behavior is shown in Eq. (2):

$$A_i = \frac{\sum_{j=1}^N V_j}{N} \quad (2)$$

where V_j represents the speed of the j th neighbor.

- *Cohesion* indicates how individuals fly from the neighborhood area to the center of mass. It refers to the tendency of individuals to fly towards the neighboring center of mass. The mathematical modeling of the Cohesion behavior is presented in Eq. (3).

$$C_i = \frac{\sum_{j=1}^N X_j}{N} - X \quad (3)$$

- *Attraction* represents how the food source attracts the individuals that fly towards it. The mathematical modeling of this behavior is shown in Eq. (4).

$$F_i = F_{loc} - X \quad (4)$$

where F_{loc} represents the position of the food source.

- *Distraction* refers to the tendency of individuals to fly away from an enemy. The distraction between the i th solution and the enemy is mathematically modeled in Eq. (5).

$$E_i = E_{loc} + X \quad (5)$$

where E_{loc} symbolize the enemy's position.

During the search process in DA, the fitness of the food source and the location are updated using the candidate with the best fitness. Furthermore, the worst candidate updates the fitness and the location of the enemy. This results

in the divergence into favorable search areas and moving away from unfavorable search areas.

The generic framework of the PSO algorithm is used by DA as it uses two vectors to update the position of a dragonfly: the step vector (ΔX) that is similar to the PSO velocity vector and the position vector.

The step vector (modeled in Eq. (6)) serves to alter the dragonflies' movement.

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \quad (6)$$

where s , a , c , f , and e are the weights of the separation S_i , alignment A_i , cohesion C_i , movement speed into the food source F_i , and the enemy disturbance level E_i of the i th individual respectively. Eq. (7) shows how these parameters are adaptively tuned during the optimization process to maintain a good balance between exploration and exploitation. Note that w is the inertia weight that is calculated based on Eq. (8). More details about the values of these parameters and their effect on the DA behavior can be found in [54].

$$\begin{aligned} s &= 2 \times r \times pct \\ a &= 2 \times r \times pct \\ c &= 2 \times r \times pct \\ f &= 2 \times r \\ e &= pct; \end{aligned} \quad (7)$$

$$w = 0.9 - Iter * \frac{(0.9 - 0.4)}{Max_iter} \quad (8)$$

where pct is calculated as in Eq. (9)

$$pct = \begin{cases} 0.1 - \frac{0.2 \times iter}{max_iter}, & \text{if } (2 \times Iter) \leq Max_Iter \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where r is a random number in the interval of $[0,1]$.

The position of an individual is updated as in Eq. (10):

$$X_{t+1} = X_t + \Delta X_{t+1} \quad (10)$$

where t is the present step.

Algorithm 1 shows the pseudo-code of the DA. Initially, the algorithm creates a randomly generated population and initialize it with step vectors randomly. Iteratively, the algorithm executes the following steps until a termination criterion is met. First, a fitness function is used to evaluate each individual in the population. Second, the algorithm updates the main coefficients (i.e., s , w , a , c , f , and e). Third, the operators: separation (S), alignment (A), cohesion (C), food source (F), and enemy (E) are altered using Eqs. (1) to (5). Finally, the Eqs. (6) and (10) are used to update the step vectors and the dragonfly position. Consequently, the best solution obtained so far is returned.

Algorithm 1 Pseudocode of the DA

```

Initialize the population  $X_i (i = 1, 2, \dots, n)$ 
Initialize  $\Delta X_i (i = 1, 2, \dots, n)$ 
while (end condition is not satisfied) do
    Evaluate each dragonfly
    Update ( $F$ ) and ( $E$ )
    Update the main coefficients ( $i$ ,  $e$ ,  $w$ ,  $s$ ,  $a$ ,  $c$ ,  $f$ , and  $e$ )
    Calculate  $S$ ,  $A$ ,  $C$ ,  $F$ , and  $E$  (using Eqs. (1) to (5))
    Update step vectors ( $\Delta X_{t+1}$ ) using Eq. (6)
    Update  $X_{t+1}$  using Eq. (10)
Return the best solution
  
```

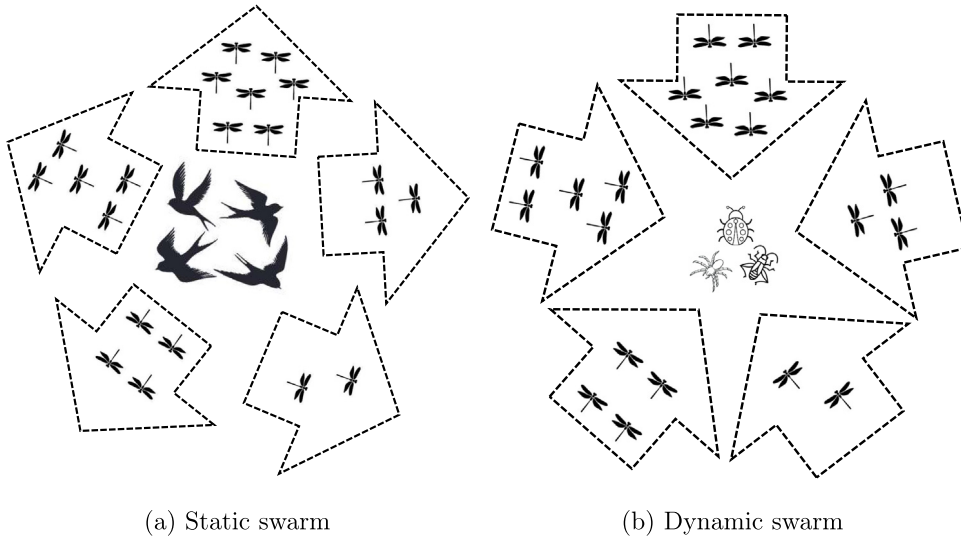


Fig. 1. Hunting and feeding swarming behavior of dragonflies when foraging.

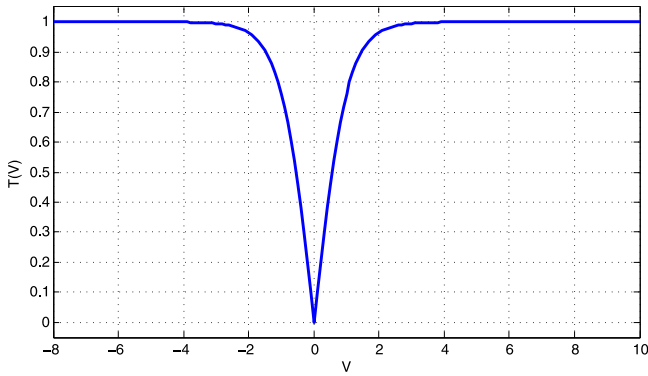


Fig. 2. V-shaped transfer function.

3. Binary Dragonfly Algorithm (BDA) for Feature Selection

FS optimization is considered a binary optimization problem. The search space is shaped as a hypercube, where an individual location is identified in the search space using the position vector $x = \{x_1, x_2, \dots, x_d\}$. DA is originally proposed to handle continuous optimization problems. The individual position is updated by adding the current position vector to the step vector. This technique needs to be changed to handle binary optimization problems. V-shaped transfer function is used to convert the continuous values into binary which is drawn as in Fig. 2 [73].

Using TFs, the positions are converted from continuous to binary using two steps. First, the value of the d th dimension of the i th step vector (velocity) in the current iteration (t) is used as an input to Eq. (11) to generate the probability of changing that element to 0 or 1. Second, set the element's value to 0 or 1 based on Eq. (12). The use of the TF in Eq. (11) is based on a previous recommendation from the literature [54].

$$T(v_d^i(t)) = |(v_d^i(t)) / \sqrt{1 + (v_d^i(t))^2}| \quad (11)$$

The i th element of the position vector is converted to 0 or 1 using Eq. (12) by plugging the outcome $T(v_k^i(t))$ obtained from

Eq. (11).

$$X(t+1) = \begin{cases} -X_t & r < T(v_k^i(t)) \\ X_t & r \geq T(v_k^i(t)) \end{cases} \quad (12)$$

where r is a function that generates a random number between 0 and 1. The value of r has a major role in deciding whether the value of X_t is flipped. When the value of $T(v_k^i(t))$ is small, the chance of flipping the new value $X(t+1)$ will be also small.

3.1. Objective function

The objective function is a key aspect that should be considered when designing any optimization problem. The wrapper FS approaches aim to reduce the number of features to the minimum and increase the learning algorithm accuracy. These two contradictory objectives should be considered in the objective function. In this work, an objective function that considers the selection ratio (minimization) and the classification error rate (minimization) is applied as shown in Eq. (13).

$$Fitness = \alpha \times ERR(D) + \beta \times \frac{|R|}{|N|} \quad (13)$$

where $ERR(D)$ represents the classification error rate (obtained by using the k-Nearest Neighbor (k-NN) classifier), $|R|$ is the number of selected features and $|N|$ is the original number of features, α and β are two parameters in the interval $[0, 1]$, where α is the complement of β [74]. These two parameters represent the weights of the classification error rate and the selection ratio respectively.

4. The proposed approach

In the wrapper-based FS methods, finding the optimal combination of features is a challenging problem since the aim is to find the subset with the minimum number of features that reveal the maximum accuracy. Moreover, FS has been considered an NP-hard problem [75]. Therefore, a high-performance meta-heuristic algorithm is required to reduce the processing time of such problems. As mentioned in the literature [76,77], the DA algorithm shows high performance when tackling various optimization problems. The merits of DA encouraged us to propose a novel approach based on the DA optimizer, and use it as a searching strategy in a FS method.

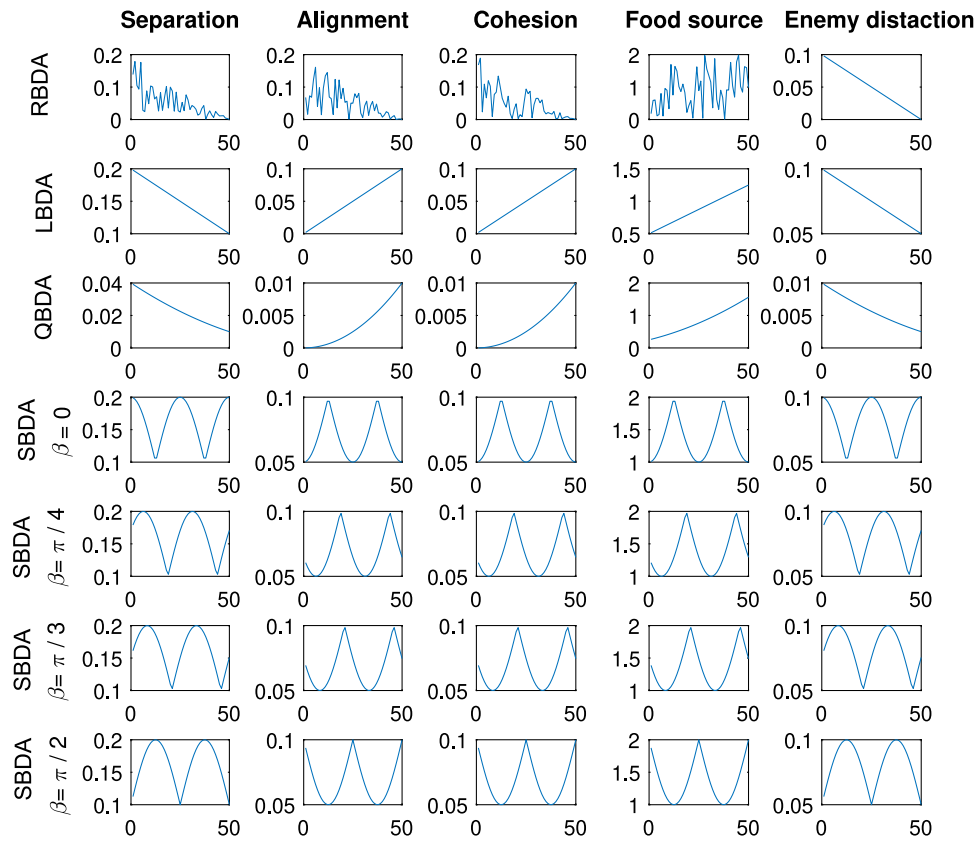


Fig. 3. Models.

According to [78], the two major factors that highly influence the performance of any population-based metaheuristic algorithm are the exploration (diversification) and the exploitation (intensification). Exploration means to diversify the solutions in the population, so the search space is explored globally. While exploitation means to focus on the neighbor area of a current good solution. Exploration via randomization helps the solutions to avoid trapping at the local optima and increases the population diversity. On the other hand, exploitation helps the searching process to converge into an optimal solution. Thus, having a good balance between these components ensures that the global optimality is achievable.

In the early stages of the optimization process, it is important to explore the different regions of the search space. Thus, the exploration is needed in this stage. However, the need for discovering the best solutions in the explored regions becomes mandatory in the last stages of the search process, so exploitation is the key in these stages.

As a aforementioned in the original DF algorithm, the coefficient s , a , c , f , and e represent the weights of the separation S_i , alignment A_i , cohesion C_i , movement speed into the food source F_i , and enemy disturbance level E_i of the i th individual respectively. The operator a is used to determine the size of the portion that should be taken from the current agents to generate the new agents. The value of 'a' determines the size of the portion based on the alignment operation presented in Eq. (7). In the original DF, the value of 'a' is randomly updated during the search starting from 2 degraded to 0.

In this paper, three versions of BDA are proposed with various updating strategies for the coefficients which affect the search. These versions utilize three different migration models to improve the search along with the original DA. These models include the linear random migration model, quadratic migration model,

and sinusoidal migration model. The curves of these models are shown in Fig. 3, where the original curve, linear migration curve, quadratic migration curve, sinusoidal migration curve are presented.

4.1. Random based Binary Dragonfly (RBDA)

In the Random based Binary DragonFly (RBDF), the values of the coefficients s , a , c , f , and e are updated as shown in Eq. (7). This model depends on the random value of r and pct . In Fig. 3. The curve of the original RBDF shows that the trends of the coefficients non-linearly fluctuate. This affects the search behavior of BDA which can turn into exploration or exploitation without rigorous structure.

4.2. Linear based Binary Dragonfly (LBDA)

A linear model is used in LBDA as shown in Eq. (14). The values of the coefficients are linearly updated each with a specific range. For example, the value range of s coefficient is updated from 0.2 to 0. The value range of e coefficient is updated from 0.1 to 0. The value ranges of a and c coefficients are updated from 0 to 0.2. The value range of f coefficient is updated from 0.0 to 2.0. These values are selected as suggested Mirjalili [54], Seyedali Mirjalili,¹ and dragonfly algorithm toolbox by Mathworks.²

$$\begin{aligned} s &= 0.2 - (0.2 \times \text{iter}/\text{max_iter}) \\ e &= 0.1 - (0.1 \times \text{iter}/\text{max_iter}) \\ a &= 0.0 + (0.2 \times \text{iter}/\text{max_iter}) \\ c &= 0.0 + (0.2 \times \text{iter}/\text{max_iter}) \\ f &= 0.0 + (2.0 \times \text{iter}/\text{max_iter}) \end{aligned} \quad (14)$$

¹ <http://www.alimirjalili.com/DA.html>.

² <https://www.mathworks.com/matlabcentral/fileexchange/51031-dragonfly-algorithm-toolbox>.

In Fig. 3, the curve of the LBDA shows that the trends of the coefficients are linearly increased or decreased. This affects the search behavior of BDA which can be turned towards exploration or exploitation with linear structure and may lead to local optima.

4.3. Quadratic based Binary Dragonfly (QBDA)

The quadratic model is used in QBDA as shown in Eq. (15). The values of the coefficients are updated in a non-linear structure. The used coefficient ranges are the same as LBDA. In Fig. 3, the curve of the QBDF shows that the trends of the coefficients are non-linearly increased or decreased. This affects the search behavior of BDA which can be turned towards exploration or exploitation in a faster base when compared to the linear structure.

$$\begin{aligned} s &= 0.2 - (0.2 * iter / max_iter)^2 \\ e &= 0.1 - (0.1 * iter / max_iter)^2 \\ a &= 0.0 + (0.2 * iter / max_iter)^2 \\ c &= 0.0 + (0.2 * iter / max_iter)^2 \\ f &= 0.0 + (2.0 * iter / max_iter)^2 \end{aligned} \quad (15)$$

It should be noted that when *iter* is small, the value of such coefficients is rapidly decreasing/increasing from its maximum, while the value of such coefficients slowly increases from zero. When *iter* is large, the value of such coefficients gradually decreases/increases to zero, and the value of such coefficients rapidly increases to its maximum.

4.4. Sinusoidal based Binary Dragonfly (SBDA)

The Sinusoidal model is used in SBDA as shown in Eq. (16). The values of the coefficients are updated in a non-linear structure. The used coefficient ranges are the same as LBDA.

$$\begin{aligned} s &= 0.10 + (0.10 \times |\cos(iter / max_iter / 4 \times \pi - \beta)|) \\ e &= 0.05 + (0.05 \times |\cos(iter / max_iter / 4 \times \pi - \beta)|) \\ a &= 0.10 - (0.05 \times |\cos(iter / max_iter / 4 \times \pi - \beta)|) \\ c &= 0.10 - (0.05 \times |\cos(iter / max_iter / 4 \times \pi - \beta)|) \\ f &= 2.00 - (1.00 \times |\cos(iter / max_iter / 4 \times \pi - \beta)|) \end{aligned} \quad (16)$$

In Fig. 3, the curve of the SBDA shows that the trends of the coefficients are non-linearly increased gradually, and when it reaches the maximum value of the coefficients then it will start decreasing gradually. The curves of this model look like sinusoidal waves. This affects the search behavior of BDA which can turn towards exploration or exploitation with wavelet structure as can be noted in the curve, where β represents the trigonometric offset angle. Typically the value of β is set to be between 0 and $\pi/2$. In this research, the parameter β in Eq. (16) takes the values 0, $\pi/4$, $\pi/3$, and $\pi/2$ to investigate its effect on the performance of the SBDA.

5. Experiments and results discussion

The performance of the proposed BDA-based algorithms (i.e., RBDA, LBDA, QBDA, and SBDA) on solving the feature selection problem is studied in this section. To elaborate, the characteristics of the dataset are summarized in Section 5.1. The performance of the proposed approaches is analyzed and discussed in Section 5.3. Finally, the best results of the proposed approaches are compared against other comparative approaches in Section 5.4.

Table 1 illustrates the settings of all parameters of the proposed approaches. These settings are selected based on extensive experiments. Due to the stochastic nature of metaheuristics algorithms 30 independent runs were repeated for each approach. Furthermore, MATLAB 2014a is used to implement all approaches. A PC with Intel Core i7, 2.2 GHz CPU and 8 GB of RAM was used to run all experiments.

Table 1

The parameter settings.

Parameter	Value
Population size	10
Number of iterations	100
Dimension	Number of features
Number of runs for each technique	30
α in fitness function	0.99
β in fitness function	0.01

Table 2

Details of datasets.

Dataset	No. of attributes	No. of objects	No. of classes
Breastcancer	9	699	2
BreastEW	30	569	2
Exactly	13	1000	2
Exactly2	13	1000	2
HeartEW	13	270	2
Lymphography	18	148	4
M-of-n	13	1000	2
PenglungEW	325	73	7
SonarEW	60	208	2
SpectEW	22	267	2
CongressEW	16	435	2
IonosphereEW	34	351	2
KrvskpEW	36	3196	2
Tic-tac-toe	9	958	2
Vote	16	300	2
WaveformEW	40	5000	3
WineEW	13	178	3
Zoo	16	101	7

5.1. The datasets characteristics

The performance of the proposed approaches is evaluated using eighteen well-known datasets from UCI repository [79]. More details about the used datasets can be found in Table 2. Each dataset is divided randomly into two sets: 80% for training and the remaining 20% is used for testing [29].

5.2. Effect of the parameter β

In this section, the effect of the parameter β on the behavior of SBDA algorithm is investigated using four different values (i.e., $\beta=0$, $\beta = \pi/4$, $\beta = \pi/3$, and $\beta = \pi/2$). Table 3 shows the behavior of the proposed SBDA algorithm with different settings of the parameter β in terms of classification accuracy. In this table, the best results are highlighted using a bold font. The highest is the best. The results show that SBDA algorithm obtained the optimal results on four datasets (Exactly, M-of-n, WineEW, and Zoo) using the four different settings of the parameter β . In other words, the results indicate that no influence of the different settings of the parameter β on the behavior of SBDA on four of the datasets. On the other hand, the SBDA with $\beta=0$ achieved the best results on four datasets. While the SBDA with $\beta = \pi/4$ obtained the best results on two datasets. The SBDA with $\beta = \pi/3$ got the best results on seven datasets, and the SBDA with $\beta = \pi/2$ obtained the best results on one dataset. In conclusion, the performance of SBDA with $\beta = \pi/3$ outperforms the other three versions of SBDA algorithm. Hence, the results of this version will be used in the upcoming experiments.

The influence of the parameter β on the behavior of the SBDA algorithm in terms of the selected features is recorded in Table 4. Again, the best results are highlighted in bold. As shown in this table, the SBDA algorithm with $\beta=0$ obtains the minimum average of the selected features on 7 datasets. While the SBDA algorithm with $\beta = \pi/4$ and $\beta = \pi/3$ obtain the minimum average of the selected features on 5 datasets. Finally, the SBDA algorithm with

Table 3

The classification accuracy of study the effect of the parameter β on the behavior of SBDA algorithm.

Dataset	Matric	$\beta = 0$	$\beta = \pi/4$	$\beta = \pi/3$	$\beta = \pi/2$
Breastcancer	Avg StDev	0.985 0.002	0.979 0.000	0.993 0.000	0.992 0.002
BreastEW	Avg StDev	0.995 0.005	0.974 0.006	0.975 0.006	0.983 0.006
Exactly	Avg StDev	1.000 0.000	1.000 0.000	1.000 0.000	1.000 0.000
Exactly2	Avg StDev	0.815 0.001	0.748 0.006	0.757 0.014	0.749 0.017
HeartEW	Avg StDev	0.876 0.009	0.891 0.011	0.867 0.009	0.848 0.017
Lymphography	Avg StDev	0.922 0.020	0.953 0.019	0.954 0.016	0.955 0.028
M-of-n	Avg StDev	1.000 0.000	1.000 0.000	1.000 0.000	1.000 0.000
penglungEW	Avg StDev	0.938 0.017	0.882 0.029	1.000 0.000	0.816 0.029
SonarEW	Avg StDev	0.968 0.019	0.981 0.018	0.993 0.011	0.968 0.018
SpectEW	Avg StDev	0.850 0.021	0.873 0.013	0.925 0.011	0.899 0.009
CongressEW	Avg StDev	0.995 0.007	0.987 0.004	0.975 0.005	0.982 0.006
IonosphereEW	Avg StDev	0.962 0.012	0.979 0.012	0.984 0.011	0.964 0.013
KrvskpEW	Avg StDev	0.975 0.006	0.983 0.004	0.966 0.004	0.978 0.005
Tic-tac-toe	Avg StDev	0.816 0.005	0.825 0.006	0.832 0.005	0.812 0.002
Vote	Avg StDev	0.983 0.003	0.972 0.010	0.972 0.008	0.969 0.006
WaveformEW	Avg StDev	0.761 0.010	0.764 0.010	0.776 0.011	0.754 0.008
WineEW	Avg StDev	1.000 0.000	1.000 0.000	1.000 0.000	1.000 0.000
Zoo	Avg StDev	1.000 0.000	1.000 0.000	1.000 0.000	1.000 0.000

$\beta = \pi/2$ achieves the minimum average of the selected features on 2 datasets.

The results of studying the effect of the parameter β on the behavior of the SBDA algorithm concerning the values of fitness function are summarized in Table 5. The table presents the average and the standard deviation of running the proposed SBDA algorithm 30 times. The best results are highlighted using a bold font, where the minimum is the best. It can be observed from the results in Table 5 that the performance of the SBDA algorithm with $\beta = \pi/3$ outperforms the other used β values on 10 datasets. While the performance of the SBDA algorithm with $\beta = \pi/0$ outperforms the other used β values on 7 datasets. The performance of the SBDA algorithm with $\beta = \pi/4$ obtained the best results on 4 datasets. Finally, the performance of the SBDA algorithm with $\beta = \pi/2$ gets the worst results when compared with the three other versions of the SBDA algorithm. As it achieves the best results in 3 datasets. To conclude, the SBDA algorithm with $\beta = \pi/3$ is better than the other selected β values in terms of classification accuracy and fitness function values.

5.3. Comparison between BDA-based algorithms

The proposed BDA-based approaches (i.e., RBDA, LBDA, QBDA, and SBDA) average classification accuracy and the standard deviation results over the 30 runs are presented in Table 6. It can be

Table 4

The number of the selected features of study the effect of the parameter β on the behavior of SBDA algorithm.

Dataset	Matric	$\beta = 0$	$\beta = \pi/4$	$\beta = \pi/3$	$\beta = \pi/2$
Breastcancer	Avg StDev	3.00 0.00	5.23 0.43	5.00 0.00	5.87 0.35
BreastEW	Avg StDev	10.87 1.87	12.57 3.17	12.20 2.54	11.50 1.94
Exactly	Avg StDev	6.07 0.25	6.07 0.25	6.13 0.35	6.07 0.25
Exactly2	Avg StDev	1.17 0.91	1.67 2.06	5.03 3.76	7.10 1.99
HeartEW	Avg StDev	7.23 0.97	5.60 2.13	6.03 0.96	5.83 1.93
Lymphography	Avg StDev	7.20 1.35	6.87 1.01	6.83 0.91	7.50 1.20
M-of-n	Avg StDev	6.13 0.35	6.13 0.35	6.07 0.25	6.13 0.35
penglungEW	Avg StDev	118.10 7.40	120.43 7.72	117.53 9.70	128.80 11.67
SonarEW	Avg StDev	24.60 3.25	25.63 3.44	24.33 2.52	24.50 3.93
SpectEW	Avg StDev	7.37 2.03	7.67 1.79	8.57 1.63	7.23 1.70
CongressEW	Avg StDev	5.87 1.80	3.67 0.96	5.40 1.50	6.00 1.31
IonosphereEW	Avg StDev	11.13 2.56	10.77 1.96	12.67 2.17	11.97 2.61
KrvskpEW	Avg StDev	18.87 2.43	19.67 2.50	19.57 2.43	18.90 2.20
Tic-tac-toe	Avg StDev	6.87 0.51	6.07 0.37	6.93 0.37	6.20 0.41
Vote	Avg StDev	4.33 0.66	5.37 1.54	4.00 0.98	3.57 1.04
WaveformEW	Avg StDev	20.77 2.27	21.27 2.88	21.83 2.65	21.67 3.12
WineEW	Avg StDev	3.33 0.48	5.77 0.82	4.40 1.07	4.37 1.33
Zoo	Avg StDev	3.20 1.42	3.17 0.38	1.97 0.96	2.30 0.99

observed from the results in Table 6 that the SBDA outperforms the other BDA-based algorithms on 11 datasets. However, the RBDA, LBDA, and QBDA obtain the best results on 6, 8, and 5 datasets respectively. In particular, the SBDA performs better than RBDA on 10 datasets. While the two algorithms achieved the same results on 3 datasets. The QBDA performs better than RBDA on 7 datasets. While both algorithms achieve the same results on one dataset. Finally, the LBDA performs better than RBDA on 7 datasets. While the two algorithms obtain the same results on 4 datasets. The standard deviation values in Table 6 indicate the robustness of the SBDA approach, as it has the lowest values in many datasets which indicates that the results are stable over several runs.

To assess the significance of the proposed algorithms, Wilcoxon rank-sum statistical analysis is conducted with $\alpha = 0.05$. The results show that the SBDA algorithm is considered to be the best algorithm as shown in Table 7. The values with bold font indicate that there is a significant difference between the results of the SBDA when it is compared against the other algorithms with $\alpha \leq 0.05$. Note that "NaN" indicates that the SBDA algorithm and the other algorithms have the same performance when running on that dataset.

The superior performance of the SBDA approach is achieved because of the non-linear behavior of the sinusoidal function. In the other approaches (i.e., RBDA, LBDA, and QBDA), the value of

Table 5

The fitness values of study the effect of the parameter β on the behavior of SBDA algorithm.

Dataset	Matric	$\beta = 0$	$\beta = \pi/4$	$\beta = \pi/3$	$\beta = \pi/2$
Breastcancer	Avg	0.018	0.028	0.013	0.015
	StDev	0.002	0.001	0.000	0.002
BreastEW	Avg	0.009	0.030	0.029	0.021
	StDev	0.005	0.006	0.006	0.005
Exactly	Avg	0.005	0.005	0.005	0.005
	StDev	0.000	0.000	0.000	0.000
Exactly2	Avg	0.184	0.251	0.245	0.255
	StDev	0.000	0.008	0.011	0.015
HeartEW	Avg	0.129	0.113	0.137	0.156
	StDev	0.008	0.010	0.008	0.016
Lymphography	Avg	0.082	0.050	0.049	0.048
	StDev	0.019	0.018	0.016	0.028
M-of-n	Avg	0.005	0.005	0.005	0.005
	StDev	0.000	0.000	0.000	0.000
penglungEW	Avg	0.065	0.120	0.004	0.187
	StDev	0.017	0.028	0.000	0.028
SonarEW	Avg	0.036	0.023	0.011	0.036
	StDev	0.019	0.018	0.011	0.018
SpectEW	Avg	0.152	0.130	0.079	0.104
	StDev	0.020	0.012	0.010	0.009
CongressEW	Avg	0.008	0.015	0.029	0.021
	StDev	0.006	0.004	0.004	0.006
IonosphereEW	Avg	0.041	0.024	0.020	0.039
	StDev	0.012	0.012	0.010	0.013
KrvskpEW	Avg	0.030	0.023	0.039	0.027
	StDev	0.006	0.004	0.004	0.005
Tic-tac-toe	Avg	0.190	0.181	0.175	0.194
	StDev	0.005	0.007	0.004	0.002
Vote	Avg	0.020	0.031	0.030	0.033
	StDev	0.003	0.010	0.008	0.005
WaveformEW	Avg	0.242	0.239	0.227	0.249
	StDev	0.010	0.010	0.011	0.008
WineEW	Avg	0.003	0.005	0.004	0.004
	StDev	0.000	0.001	0.001	0.001
Zoo	Avg	0.002	0.002	0.001	0.002
	StDev	0.001	0.000	0.001	0.001

Table 6

Average classification accuracy for the proposed BDA-based algorithms.

Dataset	Matric	RBDA	LBDA	QBDA	SBDA
Breastcancer	Avg	0.983	0.978	0.993	0.993
	StDev	0.004	0.002	0.001	0.000
BreastEW	Avg	1.000	0.987	0.980	0.975
	StDev	0.000	0.008	0.006	0.006
Exactly	Avg	1.000	1.000	0.994	1.000
	StDev	0.003	0.000	0.020	0.000
Exactly2	Avg	0.797	0.780	0.785	0.757
	StDev	0.015	0.002	0.000	0.014
HeartEW	Avg	0.839	0.901	0.880	0.867
	StDev	0.011	0.034	0.019	0.009
Lymphography	Avg	0.930	0.913	0.924	0.954
	StDev	0.021	0.019	0.023	0.016
M-of-n	Avg	1.000	1.000	0.999	1.000
	StDev	0.000	0.000	0.004	0.000
penglungEW	Avg	0.959	1.000	1.000	1.000
	StDev	0.039	0.000	0.000	0.000
SonarEW	Avg	0.964	0.944	0.948	0.993
	StDev	0.017	0.019	0.012	0.011
SpectEW	Avg	0.894	0.923	0.890	0.925
	StDev	0.010	0.010	0.013	0.011
CongressEW	Avg	0.976	0.999	0.993	0.975
	StDev	0.003	0.004	0.006	0.005
IonosphereEW	Avg	0.970	0.970	0.923	0.984
	StDev	0.013	0.009	0.012	0.011
KrvskpEW	Avg	0.975	0.981	0.968	0.966
	StDev	0.004	0.006	0.004	0.004
Tic-tac-toe	Avg	0.820	0.839	0.847	0.832
	StDev	0.005	0.000	0.005	0.005
Vote	Avg	0.996	0.971	0.959	0.972
	StDev	0.007	0.010	0.008	0.008
WaveformEW	Avg	0.766	0.760	0.738	0.776
	StDev	0.009	0.010	0.008	0.011
WineEW	Avg	0.991	1.000	1.000	1.000
	StDev	0.013	0.000	0.000	0.000
Zoo	Avg	1.000	1.000	1.000	1.000
	StDev	0.000	0.000	0.000	0.000

Table 7

Wilcoxon rank sum test results when comparing SBDA with other BDA-based algorithms based on results in Table 3.

Benchmark	RBDA	LBDA	QBDA
Breastcancer	5.19E-07	1.61E-01	2.71E-14
BreastEW	5.04E-13	4.99E-08	2.72E-04
Exactly	3.34E-01	NaN	5.57E-03
Exactly2	9.21E-13	9.49E-14	6.14E-14
HeartEW	6.75E-12	1.95E-02	1.71E-02
Lymphography	2.54E-06	8.56E-09	1.92E-05
M-of-n	NaN	NaN	1.61E-01
penglungEW	1.71E-07	1.97E-13	1.97E-13
SonarEW	1.16E-03	3.66E-08	4.66E-09
SpectEW	4.81E-08	5.96E-12	1.34E-05
CongressEW	3.09E-11	1.03E-11	7.52E-05
IonosphereEW	1.94E-02	2.91E-03	1.50E-11
KrvskpEW	7.87E-10	5.17E-01	2.86E-11
Tic-tac-toe	4.03E-06	3.16E-13	4.26E-12
Vote	1.83E-10	6.43E-01	1.20E-05
WaveformEW	4.33E-01	1.01E-01	2.43E-10
WineEW	6.18E-04	NaN	NaN
Zoo	NaN	NaN	NaN

the best results in both objectives (i.e., the minimum number of features and maximum classification accuracy) proves the ability of the sinusoidal function to improve the performance of the DA algorithm.

the coefficients almost decreases/increases linearly. This enables exploration to take place in the early stages of the search process. While exploitation takes place in the last stages of the search process. This behavior may cause a stagnation in the local optima. In contrast, the sinusoidal function interchange between exploration and exploitation during the optimization process. This can be clearly seen in Fig. 3 as the values of the coefficients vary between the minimum and maximum values on different waves during the iterations. This behavior obviously enhances the performance of the DA algorithm and enables it to explore the feature space and locate the best performance feature sets.

Table 8 presents the average number of selected features obtained by each one of the proposed BDA-based algorithms. Again, the best results are highlighted in bold. Note that the lower the value the better. It should be noted that the results of the SBDA are collected from Table 4. Where the value of the parameter β is set to $\pi/3$. This is because the $\beta = \pi/3$ gives better results of the SBDA algorithm in terms of classification accuracy. It can be seen from the results in Table 8 that RBDA performs better than other BDA-based algorithms. Where it obtains the best results on 8 datasets. The proposed SBDA gets the best results on 6 datasets. While LBDA obtains the best results on 5 datasets. Finally, the QBDA achieved the best results on 2 datasets. In particular, the SBDA performs better than RBDA on 8 datasets. While the two algorithms obtain the same results on one dataset. Obtaining

Table 8

Average number of the selected features for the proposed BDA-based algorithms.

Dataset	Matric	RBDA	LBDA	QBDA	SBDA
Breastcancer	Avg StDev	5.07 1.31	4.93 0.25	3.03 0.18	5 0
BreastEW	Avg StDev	9.07 1.74	11.7 1.97	13.33 2.51	12.2 2.54
Exactly	Avg StDev	6.07 0.25	6.13 0.35	7.03 0.85	6.13 0.35
Exactly2	Avg StDev	2.83 3.11	1.3 1.64	1.03 0.18	5.03 3.76
HeartEW	Avg StDev	6.13 1.25	6.4 1.28	6.33 1.06	6.03 0.96
Lymphography	Avg StDev	9.43 1.81	8.07 1.51	7.67 1.84	6.83 0.91
M-of-n	Avg StDev	6.07 0.25	6.07 0.25	6.97 0.67	6.07 0.25
penglungEW	Avg StDev	110.2 11.35	99.9 8.45	132.47 3.82	117.53 9.7
SonarEW	Avg StDev	23.1 3.06	26.53 4.03	28.3 3.62	24.33 2.52
SpectEW	Avg StDev	9.57 2.37	5.2 2.31	9.4 1.94	8.57 1.63
CongressEW	Avg StDev	6.07 1.23	6.43 0.86	6.43 1.22	5.4 1.5
IonosphereEW	Avg StDev	11 2.3	13.63 3.15	12.93 2.99	12.67 2.17
KrvskpEW	Avg StDev	18.93 2.12	18.97 2.83	20.6 2.09	19.57 2.43
Tic-tac-toe	Avg StDev	6.7 0.47	7 0	6.93 0.37	6.93 0.37
Vote	Avg StDev	4.3 0.53	4.63 1.45	6.23 1.77	4 0.98
WaveformEW	Avg StDev	21.4 3.54	21.4 2.3	21.77 2.71	21.83 2.65
WineEW	Avg StDev	7.13 1.43	3.43 0.68	4.07 0.69	4.4 1.07
Zoo	Avg StDev	3.4 0.56	4.2 0.41	4.5 0.73	1.97 0.96

Table 9 illustrates the results of the proposed BDA-based algorithms in terms of fitness values. Again, the best results are highlighted using a bold font, where the lowest is the best. The performance of the SBDA is almost similar to the LBDA algorithm as both of them obtain the best results on 8 datasets. This is considered to be the highest number of best results obtained by one algorithm. Furthermore, the RBDA and QBDA algorithms got the best results on 4 and 3 datasets respectively. In particular, LBDA performs better than RBDA on 8 datasets. While both algorithms get the same results on 2 datasets. The QBDA performs better than RBDA on 6 datasets. Finally, the SBDA performs better than RBDA on 12 datasets. While the two algorithms obtain the same results on one dataset. In addition, the SBDA has the lowest standard deviation values as presented in Table 9. In a nutshell, SBDA can be considered more robust than the other algorithms.

Fig. 4 illustrates the convergence behavior when running the proposed BDA-based algorithms on all datasets. Note that the *x*-axis represents the number of iterations, and the *y*-axis represents the values of the fitness function. Fig. 4 elaborates that the proposed BDA-based algorithms have the same convergence behavior on almost all datasets.

The average distance between the solutions stored in the population at each iteration for each of the proposed BDA-based algorithms is shown in Fig. 5. In this figure, four datasets (i.e., Breastcancer, HeartEW, Lymphography, and SonarEW) are selected to

Table 9

The average of the fitness values for the proposed BDA-based algorithms.

Dataset	Matric	RBDA	LBDA	QBDA	SBDA
Breastcancer	Avg StDev	0.023 0.002	0.028 0.001	0.011 0.002	0.013 0
BreastEW	Avg StDev	0.003 0.001	0.017 0.008	0.025 0.005	0.029 0.006
Exactly	Avg StDev	0.006 0.003	0.005 0	0.012 0.02	0.005 0
Exactly2	Avg StDev	0.204 0.018	0.219 0	0.214 0	0.245 0.011
HeartEW	Avg StDev	0.165 0.011	0.104 0.032	0.124 0.019	0.137 0.008
Lymphography	Avg StDev	0.075 0.02	0.091 0.018	0.079 0.022	0.049 0.016
M-of-n	Avg StDev	0.005 0	0.005 0	0.007 0.004	0.005 0
penglungEW	Avg StDev	0.044 0.038	0.003 0	0.004 0	0.004 0
SonarEW	Avg StDev	0.039 0.017	0.059 0.019	0.057 0.011	0.011 0.011
SpectEW	Avg StDev	0.11 0.009	0.079 0.009	0.113 0.012	0.079 0.01
CongressEW	Avg StDev	0.028 0.003	0.005 0.003	0.011 0.005	0.029 0.004
IonosphereEW	Avg StDev	0.033 0.013	0.033 0.009	0.081 0.012	0.02 0.01
KrvskpEW	Avg StDev	0.03 0.003	0.024 0.006	0.038 0.004	0.039 0.004
Tic-tac-toe	Avg StDev	0.187 0.004	0.169 0	0.16 0.005	0.175 0.004
Vote	Avg StDev	0.007 0.007	0.032 0.01	0.044 0.007	0.03 0.008
WaveformEW	Avg StDev	0.237 0.008	0.243 0.009	0.264 0.008	0.227 0.011
WineEW	Avg StDev	0.015 0.013	0.003 0.001	0.003 0.001	0.004 0.001
Zoo	Avg StDev	0.002 0	0.003 0	0.003 0	0.001 0.001

show the convergence speed of the proposed BDA-based algorithms. The *x*-axis represents the number of iterations, while the *y*-axis represents the distance based on the values of the fitness function. It can be observed from these plots that the RBDA has the smallest distance between the solutions in the population for all the four datasets. The solutions in the population are identical by the RBDA before finishing the maximum number of iterations for Breastcancer, HeartEW, and Lymphography. In other words, the RBDA has a faster convergence than the other three BDA-based algorithms and thus getting stuck in local optima. However, the highest distance among the solutions is obtained by QBDA. This is because the QBDA has a slow convergence and thus obtained the worst results.

5.4. Comparison with the other binary-based optimization algorithms

In the previous section, a comparison between the proposed BDA-based algorithms is conducted. In this section, the results of the proposed BDA-based algorithms are compared against binary versions of the following metaheuristics-based feature selection algorithms: binary grey wolf optimizer (BGWO), binary gravitational search algorithm (BGSA), and binary bat algorithm (BBA). The comparison is performed in terms of classification accuracy, the number of selected features, and the best-obtained fitness

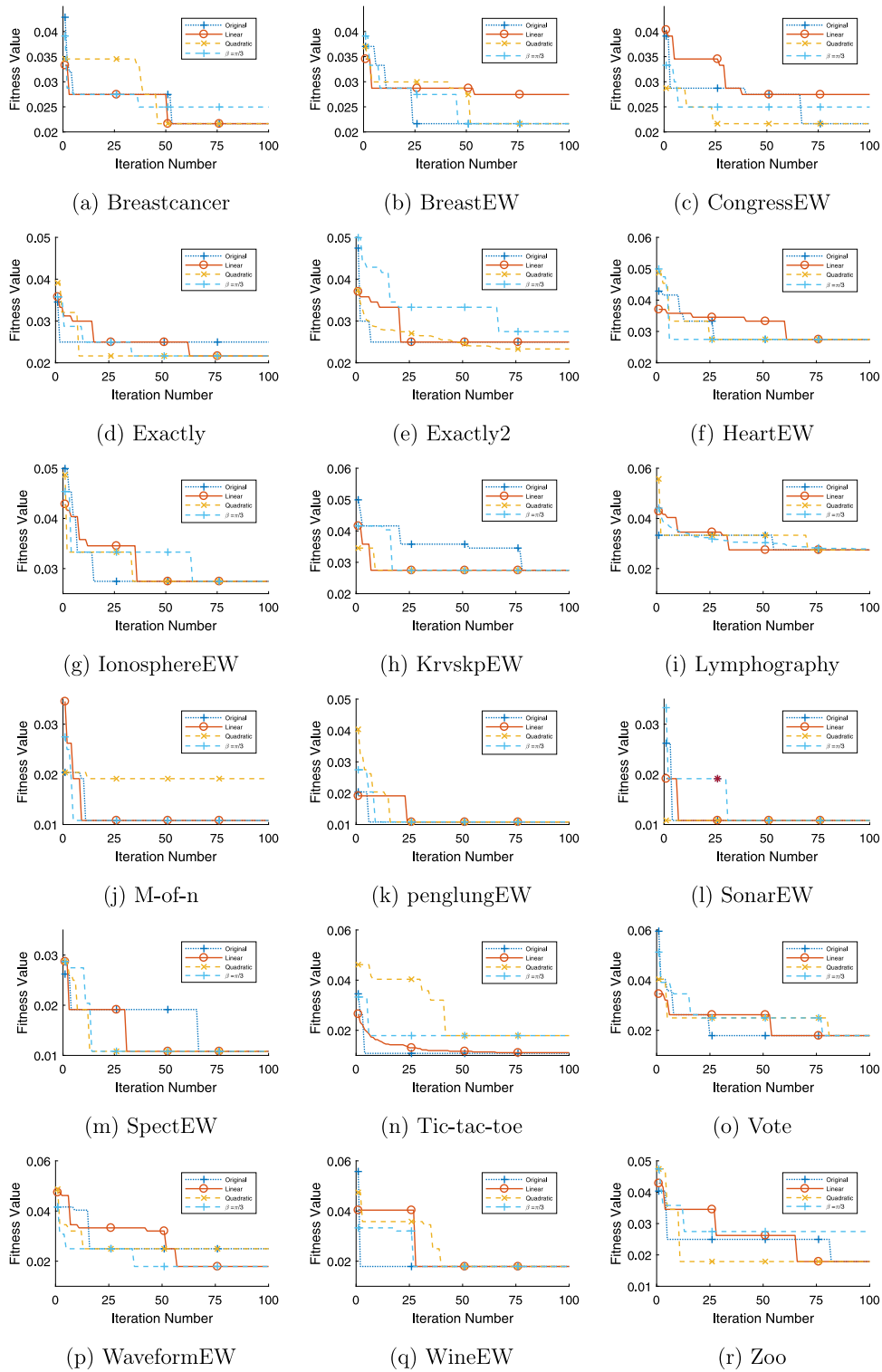


Fig. 4. Convergence behavior of the proposed BDA-based algorithms on all datasets.

values. Table 14 shows the settings of the different parameters used to run these algorithms.

Table 10 shows the average and the standard deviation of the results obtained by the proposed BDA-based algorithms, BGWO, BGSA, and BBA in terms of average classification accuracy. It can be observed that the performance of SBDA outperforms other

algorithms as it obtains the best results on 11 out of 18 datasets. The LBDA version achieves 8 out of 18 best classification accuracy. The RBDA version achieves 6 out of 18 best classification accuracy while QBDA can achieve the best classification accuracy in 5 out of 18 datasets. In brief, the proposed BDA-based algorithms can excel the comparative methods in all datasets used.

Table 10

Comparison with the other binary-based approaches in terms of average classification accuracy.

Dataset	Matric	RBDA	LBDA	QBDA	SBDA	BGWO	BGSA	BBA
Breastcancer	Avg StDev	0.983 0.004	0.978 0.002	0.993 0.001	0.993 0	0.978 0.01	0.948 0.02	0.932 0.051
BreastEW	Avg StDev	1 0	0.987 0.008	0.98 0.006	0.975 0.006	0.923 0.015	0.928 0.014	0.913 0.035
Exactly	Avg StDev	1 0.003	1 0	0.994 0.02	1 0	0.835 0.077	0.732 0.124	0.602 0.055
Exactly2	Avg StDev	0.797 0.015	0.78 0.002	0.785 0	0.757 0.014	0.674 0.041	0.644 0.041	0.683 0.04
HeartEW	Avg StDev	0.839 0.011	0.901 0.034	0.88 0.019	0.867 0.009	0.788 0.039	0.77 0.066	0.728 0.061
Lymphography	Avg StDev	0.93 0.021	0.913 0.019	0.924 0.023	0.954 0.016	0.842 0.057	0.864 0.081	0.689 0.103
M-of-n	Avg StDev	1 0	1 0	0.999 0.004	1 0	0.913 0.052	0.827 0.061	0.716 0.083
penglungEW	Avg StDev	0.959 0.039	1 0	1 0	1 0	0.869 0.012	0.949 0.054	0.816 0.054
SonarEW	Avg StDev	0.964 0.017	0.944 0.019	0.948 0.012	0.993 0.011	0.887 0.04	0.865 0.047	0.814 0.059
SpectEW	Avg StDev	0.894 0.01	0.923 0.01	0.89 0.013	0.925 0.011	0.818 0.029	0.785 0.034	0.756 0.039
CongressEW	Avg StDev	0.976 0.003	0.999 0.004	0.993 0.006	0.975 0.005	0.95 0.047	0.943 0.026	0.869 0.08
IonosphereEW	Avg StDev	0.97 0.013	0.97 0.009	0.923 0.012	0.984 0.011	0.891 0.025	0.869 0.026	0.866 0.027
KrvskpEW	Avg StDev	0.975 0.004	0.981 0.006	0.968 0.004	0.966 0.004	0.935 0.019	0.898 0.053	0.79 0.09
Tic-tac-toe	Avg StDev	0.82 0.005	0.839 0	0.847 0.005	0.832 0.005	0.806 0.029	0.761 0.038	0.658 0.081
Vote	Avg StDev	0.996 0.007	0.971 0.01	0.959 0.008	0.972 0.008	0.939 0.021	0.943 0.025	0.856 0.102
WaveformEW	Avg StDev	0.766 0.009	0.76 0.01	0.738 0.008	0.776 0.011	0.705 0.015	0.697 0.021	0.659 0.046
WineEW	Avg StDev	0.991 0.013	1 0	1 0	1 0	0.938 0.036	0.976 0.035	0.838 0.131
Zoo	Avg StDev	1 0	1 0	1 0	1 0	0.993 0.023	0.995 0.015	0.867 0.114

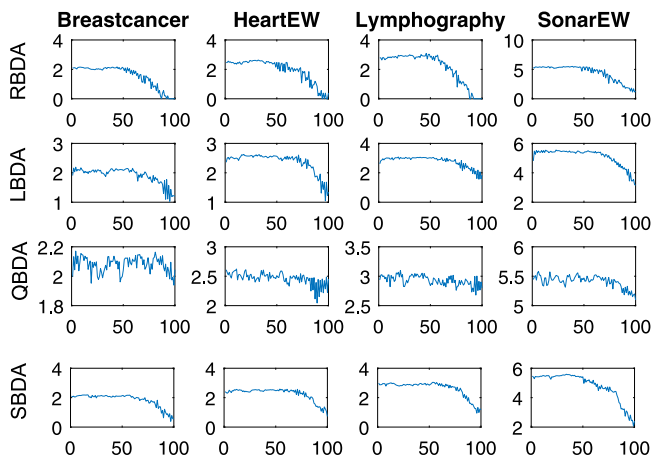
**Fig. 5.** The average distance between different solution in the population for some selected datasets.

Table 11 illustrates the results of running the four algorithms in terms of the average number of the selected features. The performance of the BBA algorithm outperforms the other comparative algorithms on 6 out of 18 datasets. The proposed SBDA

performs better than the other algorithms in 4 datasets, while LBDA and RBDA excel other comparative methods in 3 datasets each. Lastly, QBDA is able to obtain 2 new best results.

Table 12 illustrates the best fitness values obtained by the proposed BDA-based algorithms, BGWO, BGSA, and BBA algorithms. It can be observed that SBDA and LBDA show superior performance among all approaches. It obtains the best fitness values on 8 out of 18 datasets. RBDA and QBDA achieve the best results on 4 and 3 datasets respectively.

Apparently, the proposed BDA-based algorithms outperform other comparative methods in almost all datasets used in terms of classification accuracy, the average number of selected features, and the fitness function. Therefore, the performance of the proposed algorithms is superior. The ranking order of the algorithms is as follows: SBDA, LBDA, RBDA, then QBDA. Due to its superior results, it is recommended to use SBDA for feature selection problems.

5.5. Comparison with the state-of-the-art algorithms

Finally, a comparison of the proposed algorithms (RBDA, LBDA, QBDA, and SBDA) against the results of other state-of-the-art algorithms is presented in Table 13. The parameter settings of the comparative methods are provided in Table 14. Furthermore, Table 15 shows the results of the proposed algorithms when

Table 11

Comparison with the other binary-based approaches in terms of average selected features.

Dataset	Matric	RBDA	LBDA	QBDA	SBDA	BGWO	BGSA	BBA
Breastcancer	Avg	5.07	4.93	3.03	5	6.4	4.47	4.1
	StDev	1.31	0.25	0.18	0	1.75	1.01	1.27
BreastEW	Avg	9.07	11.7	13.33	12.2	21.57	14.93	11.77
	StDev	1.74	1.97	2.51	2.54	4.8	2	3.94
Exactly	Avg	6.07	6.13	7.03	6.13	10.7	7.67	5.23
	StDev	0.25	0.35	0.85	0.35	2.02	1.49	2.25
Exactly2	Avg	2.83	1.3	1.03	5.03	6.97	6.13	5.77
	StDev	3.11	1.64	0.18	3.76	2.74	2.08	1.57
HeartEW	Avg	6.13	6.4	6.33	6.03	9.7	6.63	5.07
	StDev	1.25	1.28	1.06	0.96	1.99	1.94	1.7
Lymphography	Avg	9.43	8.07	7.67	6.83	10.6	9	6.87
	StDev	1.81	1.51	1.84	0.91	2.63	2.18	1.96
M-of-n	Avg	6.07	6.07	6.97	6.07	10.43	8.2	5.73
	StDev	0.25	0.25	0.67	0.25	1.45	1.16	1.82
penglungEW	Avg	110.2	99.9	132.47	117.53	152.33	145.1	126.47
	StDev	11.35	8.45	3.82	9.7	7	4.88	15.62
SonarEW	Avg	23.1	26.53	28.3	24.33	34.87	27.07	23.53
	StDev	3.06	4.03	3.62	2.52	7.81	3.64	5.15
SpectEW	Avg	9.57	5.2	9.4	8.57	13.77	9.77	8.73
	StDev	2.37	2.31	1.94	1.63	2.93	2.3	2.29
CongressEW	Avg	6.07	6.43	6.43	5.4	10	7	5.7
	StDev	1.23	0.86	1.22	1.5	1.88	1.91	2.18
IonosphereEW	Avg	11	13.63	12.93	12.67	16.17	14.9	12.3
	StDev	2.3	3.15	2.99	2.17	2.35	2.89	3.4
KrvskpEW	Avg	18.93	18.97	20.6	19.57	30.9	19.73	14.97
	StDev	2.12	2.83	2.09	2.43	2.93	2.36	2.88
Tic-tac-toe	Avg	6.7	7	6.93	6.93	8.3	5.6	4.3
	StDev	0.47	0	0.37	0.37	1.24	0.97	1.7
Vote	Avg	4.3	4.63	6.23	4	8.63	7.37	6.1
	StDev	0.53	1.45	1.77	0.98	2.63	1.67	2.14
WaveformEW	Avg	21.4	21.4	21.77	21.83	34.07	21.6	16.23
	StDev	3.54	2.3	2.71	2.65	4.48	3.69	4.08
WineEW	Avg	7.13	3.43	4.07	4.4	7.37	6.57	4.87
	StDev	1.43	0.68	0.69	1.07	1.67	1.36	1.87
Zoo	Avg	3.4	4.2	4.5	1.97	7.37	6.97	6.43
	StDev	0.56	0.41	0.73	0.96	1.63	1.25	1.83

compared against other comparative methods in terms of classification accuracy. It should be noted that the results of the proposed algorithms are extracted from Table 6.

Table 15 shows that the proposed SBDA provides superior performance as it obtains the best new results in 4 out of 18 datasets (i.e., Lymphography, SonarEW, SpectEW, and IonosphereEW). These results are not achieved by any of the other comparative algorithms, while it achieves optimal results in 5 out of 18 datasets (i.e., Exactly, M-of-n, penglungEW, WineEW, and Zoo). LBDA obtains the best new results in 3 out of 18 datasets (i.e., HeartEW, CongressEW, and KrvskpEW), while it obtains optimal results in 5 out of 18 datasets (i.e., Exactly, M-of-n, penglungEW, WineEW, and Zoo). RBDA obtains the best new results in 3 out of 18 datasets (i.e., BreastEW, Exactly2, and Vote), while it obtains optimal results in 3 out of 18 datasets (i.e., Exactly, M-of-n, and Zoo). Furthermore, QBDA obtains the best new results in 2 out of 18 datasets (i.e., Breastcancer, and Tic-tac-toe), while it obtains optimal results in 3 out of 18 datasets (i.e., penglungEW, WineEW, and Zoo).

In summary, the proposed algorithms are able to obtain 15 out of 18 new best results, and 2 out of 18 best results similar to other comparative methods as can be shown in Table 15. It is clear that the proposed modifications on the DA algorithm has a positive impact on the search strategy of the algorithm, and thus accomplish the highest classification accuracy for almost all datasets with different dimensions.

As can be borne out by the comparative results, the SBDA excels the nine state-of-the-art methods in 12 out of 18 UCI datasets. This confirms that the SBDA version is able to improve the convergence behavior of the original BDA. The original BDA uses a random strategy to update the coefficients of BDA. This negatively affects its ability to reach the right balance between the diversification and intensification process while navigating the search space. The SBDA can overcome this major concern and structurally update the coefficients of BDA using a sinusoidal function to wave smoothly through the search space. Thus it utilizes the natural selection principle of the survival-of-the-fittest. The obtained results reveal that the use of the sinusoidal function in updating the coefficients of the BDA algorithm enhances the behavior of the original algorithm when tackling the FS problem.

6. Conclusion and future work

In this paper, the binary optimization problem Feature Selection (FS) is tackled. The Binary Dragonfly Algorithm (BDA) is modified for Feature Selection (FS) problems. The modifications for BDA update the values of five main coefficients (s , a , c , f , and e) of the BDA. Three versions of BDA are proposed and compared against the Random based Binary Dragonfly (RBDA) which is the original algorithm. The proposed versions are (1) Linear based Binary Dragonfly (LBDA), (2) Quadratic based Binary Dragonfly (QBDA), and (3) Sinusoidal based Binary Dragonfly

Table 12

Comparison with the other binary-based approaches in terms of the best fitness value.

Dataset	Matric	RBDA	LBDA	QBDA	SBDA	BGWO	BGSA	BBA
Breastcancer	Avg	0.023	0.028	0.011	0.013	0.016	0.027	0.036
	StDev	0.002	0.001	0.002	0	0.002	0.007	0.005
BreastEW	Avg	0.003	0.017	0.025	0.029	0.043	0.039	0.036
	StDev	0.001	0.008	0.005	0.006	0.007	0.01	0.009
Exactly	Avg	0.006	0.005	0.012	0.005	0.185	0.253	0.303
	StDev	0.003	0	0.02	0	0.051	0.094	0.108
Exactly2	Avg	0.204	0.219	0.214	0.245	0.249	0.288	0.25
	StDev	0.018	0	0	0.011	0.014	0.014	0.015
HeartEW	Avg	0.165	0.104	0.124	0.137	0.128	0.137	0.161
	StDev	0.011	0.032	0.019	0.008	0.026	0.03	0.023
Lymphography	Avg	0.075	0.091	0.079	0.049	0.083	0.081	0.162
	StDev	0.02	0.018	0.022	0.016	0.035	0.033	0.053
M-of-n	Avg	0.005	0.005	0.007	0.005	0.087	0.165	0.165
	StDev	0	0	0.004	0	0.039	0.041	0.044
penglungEW	Avg	0.044	0.003	0.004	0.004	0.126	0.004	0.132
	StDev	0.038	0	0	0	0.025	0	0.038
SonarEW	Avg	0.039	0.059	0.057	0.011	0.104	0.082	0.11
	StDev	0.017	0.019	0.011	0.011	0.02	0.023	0.03
SpectEW	Avg	0.11	0.079	0.113	0.079	0.143	0.153	0.143
	StDev	0.009	0.009	0.012	0.01	0.016	0.018	0.021
CongressEW	Avg	0.028	0.005	0.011	0.029	0.028	0.032	0.07
	StDev	0.003	0.003	0.005	0.004	0.01	0.013	0.015
IonosphereEW	Avg	0.033	0.033	0.081	0.020	0.099	0.127	0.124
	StDev	0.013	0.009	0.012	0.01	0.013	0.011	0.019
KrvskpEW	Avg	0.03	0.024	0.038	0.039	0.051	0.099	0.093
	StDev	0.003	0.006	0.004	0.004	0.009	0.049	0.039
Tic-tac-toe	Avg	0.187	0.169	0.160	0.175	0.177	0.232	0.232
	StDev	0.004	0	0.005	0.004	0.008	0.024	0.022
Vote	Avg	0.007	0.032	0.044	0.030	0.048	0.038	0.063
	StDev	0.007	0.01	0.007	0.008	0.009	0.009	0.017
WaveformEW	Avg	0.237	0.243	0.264	0.227	0.237	0.251	0.251
	StDev	0.008	0.009	0.008	0.011	0.008	0.013	0.016
WineEW	Avg	0.015	0.003	0.003	0.004	0.045	0.009	0.025
	StDev	0.013	0.001	0.001	0.001	0.017	0.012	0.017
Zoo	Avg	0.002	0.003	0.003	0.001	0.007	0.005	0.052
	StDev	0	0	0	0.001	0.01	0.001	0.032

(SBDA). These versions update the values of the coefficients using linear, quadratic, and sinusoidal functions, respectively.

To evaluate the proposed three versions eighteen well-known feature selection datasets from UCI repository are used. These datasets vary in terms of the number of attributes, number of objects, and number of classes. The effect of β parameter on the convergence attributes of SBDA is studied using four different values (i.e., $\beta = 0$, $\beta = \frac{\pi}{2}$, $\beta = \frac{\pi}{3}$, $\beta = \frac{\pi}{4}$). In conclusion, the SBDA with $\beta = \frac{\pi}{3}$ obtains the best results for FS. The results of four versions (i.e., RBDA, LBDA, QBDA, and SBDA with $\beta = \frac{\pi}{3}$) are also compared against each other. Interestingly, the SBDA with $\beta = \frac{\pi}{3}$ achieves the best results. In a nutshell, the values of the five main coefficients directly affect the performance of BDA.

Furthermore, the proposed version of SBDA with $\beta = \frac{\pi}{3}$ is compared against other binary swarm-based methods found in the literature (i.e., BGWO, BGSA, and BBA) using the same UCI datasets. The proposed version of SBDA with $\beta = \frac{\pi}{3}$ outperforms other swarm-based methods in all datasets in terms of average classification accuracy, 8 out of 18 datasets in terms of average selected features, and 16 out of 18 in terms of fitness function values. Furthermore, the proposed SBDA with $\beta = \frac{\pi}{3}$ is compared against nine state-of-the-art methods, and it can excel them in 12

Table 13

Key to comparative methods.

Key	Method name	Reference
BGOA	binary grasshopper optimization algorithm	[80]
BGWO	binary gray wolf optimizer	[80]
BGSA	binary gravitational search algorithm	[80]
BBA	binary bat algorithm	[80]
BSSA	binary salp swarm algorithm	[81]
HGSA	hybrid gravitational search algorithm	[35]
WOA	whale optimization algorithm	[50]
GA1	genetic algorithm	[82]
GA2	genetic algorithm	[83]
PSO1	particle swarm optimization	[82]
PSO2	particle swarm optimization	[83]

out of 18 datasets. In conclusion, the proposed SBDA with $\beta = \frac{\pi}{3}$ is a competitive method for binary optimization problems.

As the proposed algorithm shows great performance in tackling the FS problem. The proposed method can be further improved by testing different transfer functions. Furthermore, to validate the performance of the proposed BDA versions, other binary optimization problems can be investigated. Finally, other update mechanisms based on adaptive parameters can be studied in the future.

Table 14

The parameter settings of the comparative algorithms.

Algorithm	Parameter settings
BGOA	The population size = 10, and Number of iterations = 100
BGWO	The population size = 10, Number of iterations = 100, and $a = [2, 0]$
BGSA	The population size = 10, Number of iterations = 100, $G_0 = 100$, and $\alpha = 20$
BBA	The population size = 10, Number of iterations = 100, Frequency minimum $Q_{min} = 0$, Frequency maximum $Q_{max} = 2$, Loudness $A = 0.5$, and Pulse rate $r = 0.5$
BSSA	The population size = 10, and Number of iterations = 100
HGSA	The population size = 10, Number of iterations = 100, and $G_0 = 10$
WOA	The population size = 10, and Number of iterations = 100
GA1	The population size = 10, and Number of iterations = 100, the crossover with probability = 0.9, and the mutation probability = 0.01
GA2	The population size = 10, and Number of iterations = 100, the crossover with probability = 0.8, and the mutation probability = 0.02
PSO1	The population size = 10, and Number of iterations = 100, $w = 1$, and $c_1 = c_2 = 2$
PSO2	The population size = 10, and Number of iterations = 100, $w = 0.1$, and $c_1 = c_2 = 2$

Table 15

Comparison with the state-of-the-art algorithms from literature in terms of average classification accuracy.

Dataset	Matric	RBDA	LBDA	QBDA	SBDA	BGOA	BGWO	BGSA	BBA	BSSA	HGSA	WOA	GA1	PSO1	GA2	PSO2
Breastcancer	Avg StDev	0.983 0.004	0.978 0.002	0.993 0.001	0.993 0	0.98 0.001	0.968 0.002	0.957 0.004	0.937 0.031	N/A N/A	0.974 0.001	0.968 N/A	0.957 N/A	0.949 N/A	0.972 0.053	0.963 0.005
BreastEW	Avg StDev	1 0	0.987 0.008	0.98 0.006	0.975 0.006	0.947 0.005	0.954 0.007	0.942 0.006	0.931 0.014	N/A N/A	0.971 0.003	0.971 N/A	0.923 N/A	0.933 N/A	0.966 0.021	0.966 0.007
Exactly	Avg StDev	1 0.003	1 0	0.994 0.02	1 0	0.999 0.005	0.809 0.076	0.697 0.06	0.61 0.065	0.997 0.004	1 0	1 N/A	0.822 N/A	0.973 N/A	1 0.046	1 0.086
Exactly2	Avg StDev	0.797 0.015	0.78 0.002	0.785 0	0.757 0.014	0.78 0	0.743 0.017	0.706 0.023	0.628 0.057	0.767 0.012	0.77 0.006	0.742 N/A	0.677 N/A	0.666 N/A	0.757 0.042	0.768 0.027
HeartEW	Avg StDev	0.839 0.011	0.901 0.034	0.88 0.019	0.867 0.009	0.833 0.004	0.792 0.017	0.777 0.022	0.754 0.033	0.833 0.007	0.856 0.002	0.807 N/A	0.732 N/A	0.745 N/A	0.844 0.056	0.825 0.022
Lymphography	Avg StDev	0.93 0.021	0.913 0.019	0.924 0.023	0.954 0.016	0.868 0.011	0.813 0.028	0.781 0.022	0.701 0.069	0.844 0.009	0.892 0.027	0.852 N/A	0.758 N/A	0.759 N/A	0.823 0.072	0.878 0.023
M-of-n	Avg StDev	1 0	1 0	0.999 0.004	1 0	1 0	0.894 0.041	0.835 0.063	0.722 0.08	0.999 0.003	1 0	0.991 N/A	0.916 N/A	0.996 N/A	1 0.102	1 0.054
penglungEW	Avg StDev	0.959 0.039	1 0	1 0	1 0	0.927 0.013	0.85 0.014	0.919 0	0.795 0.029	0.907 0.017	0.956 0.017	0.792 N/A	0.672 N/A	0.879 N/A	0.889 0.027	0.881 0.019
SonarEW	Avg StDev	0.964 0.017	0.944 0.019	0.948 0.012	0.993 0.011	0.912 0.009	0.836 0.016	0.888 0.015	0.844 0.036	0.948 0.006	0.958 0.011	0.919 N/A	0.833 N/A	0.804 N/A	0.936 0.044	0.923 0.011
SpectEW	Avg StDev	0.894 0.01	0.923 0.01	0.89 0.013	0.925 0.011	0.826 0.01	0.81 0.014	0.783 0.024	0.8 0.027	0.833 0.007	0.919 0.006	0.866 N/A	0.756 N/A	0.738 N/A	0.872 0.038	0.874 0.011
CongressEW	Avg StDev	0.976 0.003	0.999 0.004	0.993 0.006	0.975 0.005	0.964 0.005	0.948 0.011	0.951 0.008	0.872 0.075	0.970 0.005	0.966 0.001	0.956 N/A	0.898 N/A	0.937 N/A	0.959 0.076	0.961 0.009
IonosphereEW	Avg StDev	0.97 0.013	0.97 0.009	0.923 0.012	0.984 0.011	0.899 0.007	0.885 0.009	0.881 0.01	0.877 0.019	0.938 0.005	0.934 0.007	0.926 N/A	0.863 N/A	0.876 N/A	0.917 0.028	0.931 0.015
KrvskpEW	Avg StDev	0.975 0.004	0.981 0.006	0.968 0.004	0.966 0.004	0.968 0.003	0.934 0.015	0.908 0.048	0.816 0.081	0.969 0.005	0.978 0.005	0.972 N/A	0.94 N/A	0.949 N/A	0.980 0.074	0.969 0.048
Tic-tac-toe	Avg StDev	0.82 0.005	0.839 0	0.847 0.005	0.832 0.005	0.808 0	0.754 0.032	0.753 0.024	0.665 0.063	0.797 0.000	0.788 0.008	0.785 N/A	0.764 N/A	0.75 N/A	0.794 0.059	0.799 0.021
Vote	Avg StDev	0.996 0.007	0.971 0.01	0.959 0.008	0.972 0.008	0.966 0.003	0.944 0.01	0.931 0.011	0.851 0.096	0.955 0.004	0.973 0.003	0.939 N/A	0.808 N/A	0.888 N/A	0.958 0.079	0.951 0.009
WaveformEW	Avg StDev	0.766 0.009	0.76 0.01	0.738 0.008	0.776 0.011	0.737 0.003	0.723 0.007	0.695 0.014	0.669 0.033	0.736 0.004	0.815 0.003	0.753 N/A	0.712 N/A	0.732 N/A	0.769 0.035	0.749 0.014
WineEW	Avg StDev	0.991 0.013	1 0	1 0	1 0	0.989 0	0.96 0.012	0.951 0.015	0.919 0.052	0.998 0.005	0.989 0.006	0.959 N/A	0.947 N/A	0.937 N/A	0.982 0.080	0.973 0.010
Zoo	Avg StDev	1 0	1 0	1 0	1 0	0.993 0.009	0.975 0.009	0.939 0.008	0.874 0.095	0.993 0.010	0.932 0.006	0.98 N/A	0.946 N/A	0.963 N/A	0.886 0.110	0.960 0.019

CRedit authorship contribution statement

Abdelaziz I. Hammouri: Project administration, Supervision, Conceptualization, Methodology, Writing - original draft, Software. **Majdi Mafarja:** Conceptualization, Methodology, Writing original draft. **Mohammed Azmi Al-Betar:** Writing - original draft, Writing - review & editing, Validation. **Mohammed A. Awadallah:** Writing - original draft, Data curation, Validation,

Formal analysis, Visualization. **Iyad Abu-Doush:** Data curation, Validation, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery in databases, *AI Mag.* 17 (3) (1996) 37.
- [2] J. Han, J. Pei, M. Kamber, *Data Mining: Concepts and Techniques*, Elsevier, 2011.
- [3] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, *J. Mach. Learn. Res.* 3 (Mar) (2003) 1157–1182.
- [4] Q. Al-Tashi, H.M. Rais, S.J. Abdulkadir, S. Mirjalili, H. Alhussian, A review of grey wolf optimizer-based feature selection methods for classification, in: *Evolutionary Machine Learning Techniques*, Springer, 2020, pp. 273–286.
- [5] J. Hua, W.D. Tembe, E.R. Dougherty, Performance of feature-selection methods in the classification of high-dimension data, *Pattern Recognit.* 42 (3) (2009) 409–424.
- [6] V. Gómez-Verdejo, M. Verleysen, J. Fleury, Information-theoretic feature selection for functional data classification, *Neurocomputing* 72 (16–18) (2009) 3580–3589.
- [7] R.Z. Al-Abdallah, A.S. Jaradat, I.A. Doush, Y.A. Jaradat, Abinary classifier based on firefly algorithm, *Jordanian J. Comput. Inf. Technol.* 3 (3) (2017).
- [8] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, *IEEE Trans. Knowl. Data Eng.* (4) (2005) 491–502.
- [9] S. Boutemedjet, N. Bouguila, D. Ziou, A hybrid feature extraction selection approach for high-dimensional non-Gaussian data clustering, *IEEE Trans. Pattern Anal. Mach. Intell.* 31 (8) (2008) 1429–1443.
- [10] A.I. Hammouri, S. Abdullah, Biogeography-based optimisation for data clustering, in: *13th International Conference on New Trends in Intelligent Software Methodology Tools, and Techniques, SoMet 2014*, IOS Press, 2014, pp. 951–963.
- [11] K. Huang, S. Aviyente, Wavelet feature selection for image classification, *IEEE Trans. Image Process.* 17 (9) (2008) 1709–1720.
- [12] B. Chen, L. Chen, Y. Chen, Efficient ant colony optimization for image feature selection, *Signal Process.* 93 (6) (2013) 1566–1576.
- [13] I.A. Doush, A. Sawalha, Automatic music composition using genetic algorithm and artificial neural networks, *Malaysian J. Comput. Sci.* 33 (1) (2020) 35–51.
- [14] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, Z. Wang, A novel feature selection algorithm for text categorization, *Expert Syst. Appl.* 33 (1) (2007) 1–5.
- [15] Z. Zheng, X. Wu, R. Srihari, Feature selection for text categorization on imbalanced data, *ACM Sigkdd Explor. Newsl.* 6 (1) (2004) 80–89.
- [16] H. Liu, R. Setiono, Chi2: Feature Selection and Discretization of Numeric Attributes, *IEEE*, 1995, pp. 388–391.
- [17] J. Quinlan, Induction of decision trees, *Mach. Learn.* 1 (1) (1986) 81–106.
- [18] J. Quinlan, *C4.5: Programs for Machine Learning*, Morgan kaufmann, 1993.
- [19] K.K. Kandaswamy, G. Pugalethi, M.K. Hazrati, K.-U. Kalies, T. Martinetz, BLProt: prediction of bioluminescent proteins based on support vector machine and relief feature selection, *BMC Bioinformatics* 12 (1) (2011) 345.
- [20] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, *Mach. Learn.* 53 (1) (2003) 23–69.
- [21] T.T. Le, R.J. Urbanowicz, J.H. Moore, B.A. McKinney, Statistical inference relief (STIR) feature selection, *Bioinformatics* 35 (8) (2019) 1358–1365.
- [22] Z. Huang, C. Yang, X. Zhou, T. Huang, A hybrid feature selection method based on binary state transition algorithm and relief, *IEEE J. Biomed. Health Inform.* 23 (5) (2018) 1888–1898.
- [23] Z. Deng, F.-L. Chung, S. Wang, Robust relief-feature weighting, margin maximization, and fuzzy optimization, *IEEE Trans. Fuzzy Syst.* 18 (4) (2010) 726–744.
- [24] R. Kohavi, G.H. John, Wrappers for feature subset selection, *Artif. Intell.* 97 (1–2) (1997) 273–324.
- [25] E. Pashaei, N. Aydin, Binary black hole algorithm for feature selection and classification on biological data, *Appl. Soft Comput.* 56 (2017) 94–106.
- [26] H. Faris, A.-Z. Ala'M, A.A. Heidari, I. Aljarah, M. Mafarja, M.A. Hassonah, H. Fujita, An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks, *Inf. Fusion* 48 (2019) 67–83.
- [27] H. Chantar, M. Mafarja, H. Alsawalqah, A.A. Heidari, I. Aljarah, H. Faris, Feature selection using binary grey wolf optimizer with elite-based crossover for arabic text classification, *Neural Comput. Appl.* (2019) 1–20.
- [28] I. Zelinka, A survey on evolutionary algorithms dynamics and its complexity-mutual relations, past, present and future, *Swarm Evol. Comput.* 25 (2015) 2–14.
- [29] M. Mafarja, S. Mirjalili, Whale optimization approaches for wrapper feature selection, *Appl. Soft Comput.* 62 (2017) 441–453.
- [30] J. Kennedy, Swarm intelligence, in: *Handbook of Nature-Inspired and Innovative Computing*, Springer, 2006, pp. 187–219.
- [31] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: *Micro Machine and Human Science, 1995. MHS'95, Proceedings of the Sixth International Symposium on*, IEEE, 1995, pp. 39–43.
- [32] M. Dorigo, V. Maniezzo, A. Colnari, Ant system: optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man Cybern. B* 26 (1) (1996) 29–41.
- [33] H. Turabieh, M. Mafarja, X. Li, Iterated feature selection algorithms with layered recurrent neural network for software fault prediction, *Expert Syst. Appl.* 122 (2018) 27–42.
- [34] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *J. Global Optim.* 39 (3) (2007) 459–471.
- [35] M. Taradeh, M. Mafarja, A.A. Heidari, H. Faris, I. Aljarah, S. Mirjalili, H. Fujita, An evolutionary gravitational search-based feature selection, *Inform. Sci.* (2019).
- [36] Q. Al-Tashi, H. Rais, S. Jadid, Feature selection method based on grey wolf optimization for coronary artery disease classification, in: *International Conference of Reliable Information and Communication Technology*, Springer, 2018, pp. 257–266.
- [37] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.* 69 (2014) 46–61.
- [38] Q. Al-Tashi, S.J.A. Kadir, H.M. Rais, S. Mirjalili, H. Alhussian, Binary optimization using hybrid grey wolf optimization for feature selection, *IEEE Access* 7 (2019) 39496–39508.
- [39] I.A. Doush, M.A. Al-Betar, M.A. Awadallah, A.I. Hammouri, M. Ra'ed, S. ElMustafa, H. Alkhraisat, Harmony search algorithm for patient admission scheduling problem, *J. Intell. Syst.* 29 (1) (2018) 540–553.
- [40] S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Softw.* 95 (2016) 51–67.
- [41] M. Mafarja, A. Qasem, A.A. Heidari, I. Aljarah, H. Faris, S. Mirjalili, Efficient hybrid nature-inspired binary optimizers for feature selection, *Cogn. Comput.* (2019) 1–26.
- [42] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, *Adv. Eng. Softw.* (2017).
- [43] H. Faris, A.A. Heidari, A.-Z. Ala'M, M. Mafarja, I. Aljarah, M. Eshtay, S. Mirjalili, Time-varying hierarchical chains of salps with random weight networks for feature selection, *Expert Syst. Appl.* 140 (2020) 112898.
- [44] N. Neggaz, A.A. Ewees, M.A. Elaziz, M. Mafarja, Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection, *Expert Syst. Appl.* 145 (2020) 113103.
- [45] S. Mirjalili, The ant lion optimizer, *Adv. Eng. Softw.* 83 (2015) 80–98.
- [46] M. Mafarja, S. Mirjalili, Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection, *Soft Comput.* (2018).
- [47] M. Mafarja, D. Eleyan, S. Abdullah, S. Mirjalili, 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*, ACM, 2017, p. 14.
- [48] Q. Tu, X. Chen, X. Liu, Multi-strategy ensemble grey wolf optimizer and its application to feature selection, *Appl. Soft Comput.* 76 (2019) 16–30.
- [49] M. Mafarja, S. Mirjalili, Hybrid whale optimization algorithm with simulated annealing for feature selection, *Neurocomputing* 260 (2017) 302–312.
- [50] M. Mafarja, S. Mirjalili, Whale optimization approaches for wrapper feature selection, *Appl. Soft Comput.* 62 (2018) 441–453.
- [51] M. Mafarja, I. Jaber, S. Ahmed, T. Thaher, Whale optimisation algorithm for high-dimensional small-instance feature selection, *Int. J. Parallel Emergent Distrib. Syst.* (2019) 1–17.
- [52] A.G. Hussien, A.E. Hassanien, E.H. Houssein, S. Bhattacharyya, M. Amin, S-shaped binary whale optimization algorithm for feature selection, in: S. Bhattacharyya, A. Mukherjee, H. Bhaumik, S. Das, K. Yoshida (Eds.), *Recent Trends in Signal and Image Processing*, Springer Singapore, Singapore, 2019, pp. 79–87.
- [53] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.* 1 (1) (1997) 67–82.
- [54] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, *Neural Comput. Appl.* 27 (4) (2016) 1053–1073.
- [55] V. Suresh, S. Sreejith, Generation dispatch of combined solar thermal systems using dragonfly algorithm, *Computing* 99 (1) (2017) 59–80.
- [56] D. Guha, P.K. Roy, S. Banerjee, Optimal tuning of 3 degree-of-freedom proportional-integral-derivative controller for hybrid distributed power system using dragonfly algorithm, *Comput. Electr. Eng.* 72 (2018) 137–153.
- [57] K. Sureshkumar, V. Ponnusamy, Power flow management in micro grid through renewable energy sources using a hybrid modified dragonfly algorithm with bat search algorithm, *Energy* (2019).
- [58] M.-A. Díaz-Cortés, N. Ortega-Sánchez, S. Hinojosa, D. Oliva, E. Cuevas, R. Rojas, A. Demin, A multi-level thresholding method for breast thermograms analysis using dragonfly algorithm, *Infrared Phys. Technol.* 93 (2018) 346–361.
- [59] L. Xu, H. Jia, C. Lang, X. Peng, K. Sun, A novel method for multilevel color image segmentation based on dragonfly algorithm and differential evolution, *IEEE Access* 7 (2019) 19502–19538.

- [60] M. Jafari, M.H.B. Chaleshtari, Using dragonfly algorithm for optimization of orthotropic infinite plates with a quasi-triangular cut-out, *Eur. J. Mech. A Solids* 66 (2017) 1–14.
- [61] B. Babayigit, Synthesis of concentric circular antenna arrays using dragonfly algorithm, *Int. J. Electron.* 105 (5) (2018) 784–793.
- [62] Y. Yuan, L. Lv, X. Wang, X. Song, Optimization of a frame structure using the coulomb force search strategy-based dragonfly algorithm, *Eng. Optim.* (2019) 1–17.
- [63] B.S. Yildiz, A.R. Yildiz, The harris hawks optimization algorithm, salp swarm algorithm, grasshopper optimization algorithm and dragonfly algorithm for structural design optimization of vehicle components, *Mater. Test.* 61 (8) (2019) 744–748.
- [64] S. Singh, A. Ashok, M. Kumar, T.K. Rawat, et al., Optimal design of IIR filter using dragonfly algorithm, in: *Applications of Artificial Intelligence Techniques in Engineering*, Springer Singapore, Singapore, 2019, pp. 211–223.
- [65] M. Hariharan, R. Sindhu, V. Vijejan, H. Yazid, T. Nadarajaw, S. Yaacob, K. Polat, Improved binary dragonfly optimization algorithm and wavelet packet based non-linear features for infant cry classification, *Comput. Methods Programs Biomed.* 155 (2018) 39–51.
- [66] F. Aadil, W. Ahsan, Z.U. Rehman, P.A. Shah, S. Rho, I. Mehmood, Clustering algorithm for internet of vehicles (IoV) based on dragonfly optimizer (CAVDO), *J. Supercomput.* 74 (9) (2018) 4542–4567.
- [67] A.I. Hammouri, E.T.A. Samra, M.A. Al-Betar, R.M. Khalil, Z. Alasmer, M. Kanan, A dragonfly algorithm for solving traveling salesman problem, in: *2018 8th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, IEEE, 2018, pp. 136–141.
- [68] P.T. Daely, S.Y. Shin, Range based wireless node localization using dragonfly algorithm, in: *Ubiquitous and Future Networks (ICUFN)*, 2016 Eighth International Conference on, IEEE, 2016, pp. 1012–1015.
- [69] M. Abdel-Basset, Q. Luo, F. Miao, Y. Zhou, Solving 0–1 knapsack problems by binary dragonfly algorithm, in: *International Conference on Intelligent Computing*, Springer, 2017, pp. 491–502.
- [70] M. Mafarja, I. Aljarah, A.A. Heidari, H. Faris, P. Fournier-Viger, X. Li, S. Mirjalili, Binary dragonfly optimization for feature selection using time-varying transfer functions, *Knowl.-Based Syst.* (2018).
- [71] M. Mafarja, D. Eleyan, I. Jaber, A. Hammouri, S. Mirjalili, Binary dragonfly algorithm for feature selection, in: *New Trends in Computing Sciences (ICTCS)*, 2017 International Conference on, IEEE, 2017, pp. 12–17.
- [72] C.W. Reynolds, Flocks, herds and schools: A distributed behavioral model, in: *ACM SIGGRAPH Computer Graphics*, Vol. 21, ACM, 1987, pp. 25–34.
- [73] S. Mirjalili, A. Lewis, S-shaped versus V-shaped transfer functions for binary particle swarm optimization, *Swarm Evol. Comput.* 9 (2013) 1–14.
- [74] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary ant lion approaches for feature selection, *Neurocomputing* 213 (2016) 54–65.
- [75] S.K. Pal, A. Skowron, Rough-fuzzy Hybridization: A New Trend in Decision Making, Springer-Verlag New York, Inc., 1999.
- [76] M. Mafarja, A.A. Heidari, H. Faris, S. Mirjalili, I. Aljarah, Dragonfly algorithm: theory, literature review, and application in feature selection, in: *Nature-Inspired Optimizers*, Springer, 2020, pp. 47–67.
- [77] G.I. Sayed, A. Tharwat, A.E. Hassanien, Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection, *Appl. Intell.* 49 (1) (2019) 188–205.
- [78] X.-S. Yang, *Nature-inspired optimization algorithms*, Elsevier, 2014.
- [79] M. Lichman, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2013.
- [80] M. Mafarja, I. Aljarah, A.A. Heidari, A.I. Hammouri, H. Faris, A.-Z. Ala'M, S. Mirjalili, Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems, *Knowl.-Based Syst.* 145 (2018) 25–45.
- [81] I. Aljarah, M. Mafarja, A.A. Heidari, H. Faris, Y. Zhang, S. Mirjalili, Asynchronous accelerating multi-leader salp chains for feature selection, *Appl. Soft Comput.* 71 (2018) 964–979.
- [82] S. Kashef, H. Nezamabadi-pour, An advanced ACO algorithm for feature subset selection, *Neurocomputing* 147 (2015) 271–279.
- [83] M. Mafarja, I. Aljarah, H. Faris, A.I. Hammouri, A.-Z. Ala'M, S. Mirjalili, Binary grasshopper optimisation algorithm approaches for feature selection problems, *Expert Syst. Appl.* 117 (2019) 267–286.