



Binary Black Widow with Hill Climbing Algorithm for Feature Selection

Ahmed Al-saedi and Abdul-Rahman Mawlood-Yunis(✉)

Physics and Computer Science Department, Wilfrid Laurier University,
Waterloo, ON N2L3C5, Canada

alsao290@mylaurier.ca, amawloodyunis@wlu.ca

Abstract. Feature Selection (FS) is a pre-processing step in most big data processing applications. Its purpose is to remove inconsequential and redundant features from data to determine a final set of data properties that best describe the data as a whole. The FS process is an NP-hard problem. It tries to determine the optimal subset, i.e., produces all conceivable solutions to acquire only the best. In the last few years, metaheuristic algorithms (MAs) have been coined as an ideal solution for FS problems, particularly in high-dimensional data cases. This work is an extension of our previous effort in finding an effective solution to FS problems by applying a recently developed metaheuristic algorithm called the Black Widow Optimization (BWO) algorithm. We combine our previous algorithm, the Binary Black Widow Algorithm (BBWO), with a Hill-Climbing Algorithm to solve the slow convergence problem of the BBWO. The newly developed algorithm, BBWO-HCA, is tested using 28 UCI datasets and compared with six well-regarded algorithms in the domain. The test results show that the BBWO-HCA outperforms our previous BBWO solution and almost all comparable solutions tested.

Keywords: Feature Selection · Evolutionary Algorithm · Metaheuristic Algorithm · Classification · Machine Learning · Data Mining

1 Introduction

It has become a challenge for researchers and developers to cope with the explosive growth of available data, the dimensions of which are expanding daily. Feature selection (FS) is a pre-processing step in most big data processing and machine learning applications, particularly in data mining applications. It is used to remove noisy and inconsequential features to determine a subset of features that best represent and portray the data, thus, boosting the quality of the data obtained.

Classical search approaches, such as random search and complete search have been used to solve FS problems [17]. While these methods ensure the optimal

Supported by Wilfrid Laurier University research fund.

solution for small datasets, their execution is impractical for large datasets. FS is an NP-hard problem, it tries to determine the optimal subset. For example, if a dataset contains n features, then 2^n solutions must be formulated and assessed, i.e., the problem complexity is $O(\infty)$. It also requires an enormous amount of computational power and an excessive amount of time.

In the last few years, metaheuristic algorithms (MAs) have been identified as an ideal solution to FS problems, particularly in cases involving high-dimensional data. Researchers used the Simulated Annealing algorithm, Ant Colony Optimization algorithm, Particle Swarm Optimization algorithm, Genetic Algorithm, etc. to solve FS problems and have obtained valuable results. For example, see [1, 4, 5, 8, 11, 12, 18].

MAs are the most appropriate alternative method for addressing the limitations of lengthy, far-reaching searches that entail high computational cost. Despite some desirable results, however, most MAs are impeded by the limitations imposed by a *local optimum* and a disproportion between the *explorative* and *exploitative* scope of the algorithm. Moreover, each dataset has a different number of features, and no single method is the most appropriate for FS problems, i.e., one can still find room for improvements. These observations motivated this work to look for means to overcome the limitations described and develop a novel FS solution.

We selected a recent algorithm, the Black Widow Optimization algorithm (BWO) [9], to study FS problems due to its success in optimizing engineering design problems. BWO is a nature-inspired algorithm that mimics the black widow's life cycle. It is inspired by the singular mating behaviour exhibited by the black widow spider, a process that includes an exclusive stage called *cannibalism*. The BWO approach is designed to deliver rapid convergence and to avoid local optima, and, because BWO maintains equilibrium between the exploration and exploitation stages [9], a property that most MAs applied to FS problems are lacking [14], the BWO is particularly appropriate for solving several kinds of optimization problems that involve a number of local optima.

This work is an extension of our previous effort in trying to find an effective solution to FS problems by applying the BWO algorithm. In our previous effort, we modified the BWO algorithm to solve feature selection problems and developed the Binary Black Widow Optimization (BBWO) algorithm [3].

Despite the competitive results of the BBWO, its performance can be further improved by enhancing the *slow convergence* caused by the use of a population of solutions and a lack of *local exploitation*. In this article, we describe an improved version of the BBWO. We combined the BBWO with the Hill-Climbing Algorithm (HCA). The newly developed algorithm, the BBWO-HCA, is tested using 28 UCI datasets and compared with six well-regarded algorithms in the domain. The algorithms are Binary particle swarm optimization (BPSO) [10], Binary multi-verse optimization algorithm (BMVO) [2], Binary grey wolf optimizer algorithm (BGWO) [1, 7], Binary moth-flame optimization algorithm (BMFO) [19], Binary whale optimization algorithm (BWOA) [11], and Binary bat algorithm (BBAT) [13].

The rest of this article is organized as follows: In Sect. 2, the proposed algorithm is presented. In Sect. 3, the experiment setup, test results, and result discussion are presented. Finally, in Sect. 4, the research is concluded and some future works are identified.

2 BBWO-HCA Feature Selection Algorithm

The results presented in [3] show that the BBWO produces impressive results and, in some cases, is competitive with the best-known algorithms. The results also reveal that the BBWO performance can be further improved by enhancing the *slow convergence* due to the use of a population of solutions and a lack of *local exploitation*. The BBWO-HCA aims to increase the *exploitation* process of the BBWO by incorporating it with a local metaheuristic algorithm based on the *Hill-Climbing Algorithm* (HCA). HCA is a well-known local search algorithm. It has been tested on various problems and has shown to be an effective and efficient method that can produce sound results [6]. The BBWO-HCA's main steps, *selection*, *procreation*, and *mutation*, are described in the subsections below.

2.1 Solutions Representation

In the BBWO-HCA, each solution represents a single black widow. All possible solutions to all FS problems are envisioned in terms of the attributes of the black widow spider. In programming terms, this is equivalent to saying each spider is represented by a class and spider attributes are class instance variables, or each spider is an array and spider attributes are array values. The spider population is modeled as an N_{var} dimensional array, i.e., an array of spider objects, and the FS problem becomes an N_{var} dimensional optimization problem.

The BBWO-HCA algorithm uses binary values to represent a population of solutions (N_{pop}). In binary representation, a solution is shown by a one-dimensional array. The length of the array varies in accordance with the feature number of the original dataset. For example, if S features are contained in the dataset, the solution length is S . The cell value in the array will be '1' or '0'. The value '1' indicates that the corresponding feature is selected, whereas '0' indicates that the feature is not selected. In general, when the number of features is N_f and the population size is $|N_{pop}|$, the array size of the problem will be $N_f \times |N_{pop}|$.

2.2 Initialization

The population of solutions offered by the BBWO-HCA is randomly generated by assigning a value of either "0" or "1" to each cell of the solution. The process begins by initializing the population size and the number of features. The algorithm then arbitrarily assigns either '0' or '1' by looping through each solution in the population. This process is repeated until all solutions in the population have been initialized.

2.3 Fitness Function and Evaluation

Each solution is evaluated according to a fitness function. The function employed is shown in Eq. 1. A similar function is used by [1, 15]. The k-nearest neighbour algorithm (KNN) [16] is used in the solution evaluation, i.e., the KNN classifier determines the accuracy of the solution.

$$f = \alpha \gamma_R(D) + \beta \frac{|R|}{|C|} \quad (1)$$

In Eq. 1, $\gamma_R(D)$ represents the classification error rate of the KNN classifier, $|R|$ is the cardinality of the selected subset, $|C|$ is the total number of the original features in the dataset, and α, β are two weight parameters corresponding to the importance of classification quality and subset length, $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$. A similar approach is adapted by [7, 11, 19].

After initializing the population of solutions, we assign to each solution (widow) a fitness value, which represents the quality of the solution. The fitness value of each solution is calculated using the fitness function and is evaluated using the KNN classifier. This is because the BBWO-HCA is a wrapper-based FS approach.

2.4 Transformation Function

The positions of the search agents generated from the standard BWO are continuous values. This cannot be directly applied to our problem because it contradicts the binary nature of the FS on selection or non-selection (0 or 1).

The sigmoidal function in 2 and 3, which is considered a form of the transformation function, is used in our proposed method as a part of the reproduction process to convert any continuous values to binary equivalents. The performance of the transformation function has been investigated and adopted by many researchers, e.g., [1, 7, 19].

$$z_{s_w} = \frac{1}{1 + e^{-z_w}} \quad (2)$$

$$z_{binary} = \begin{cases} 0, & \text{if } rand < z_{s_w} \\ 1, & \text{if } rand \geq z_{s_w} \end{cases} \quad (3)$$

where each of z_{s_w} is a continuous value (feature) in the search agent for the S-shaped function, specifically in the solution w at dimension d ($w = 1, \dots, d$), and is a random number drawn from the uniform distribution $\in [0, 1]$. The z_{binary} value can be 0 or 1 depending on the value of $rand$ compared to the values of z_{s_w} , where e is a mathematical constant known as Euler's number.

2.5 Reproduction Process

BWO is inspired by Darwin's natural selection theory, which is defined as generational descent accompanied by modification where species are subtly adjusted over time and new species arise as a result.

In the BBWO-BCA algorithm, the procreation process begins and parents (in pairs) are selected randomly to perform the procreating steps by mating to bring forth the new generation. An array known as *alpha* will be generated to complete further reproduction. Offspring c_1 and c_2 will be produced by taking α with the following equation in which w_1 and w_2 are parents.

$$\begin{cases} c_1 = \alpha \times w_1 + (1 - \alpha) \times w_2 \\ c_2 = \alpha \times w_2 + (1 - \alpha) \times w_1 \end{cases} \quad (4)$$

2.6 Cannibalism Process

The BWO includes an exclusive stage, *cannibalism*. Cannibalism can be classified into three kinds: *sexual cannibalism* where the husband gets eaten by the female black widow during or after mating, *sibling-cannibalism* where the weaker siblings are eaten by the stronger siblings, and *mother cannibalism* where the mother is eaten by her strongest child. The BBWO-HCA uses this concept of cannibalism and determines the weak or strong spiders by calculating and evaluating their fitness values. The best solutions (surviving spiders) from the reproduction process will be selected and stored in population two, i.e., pop_2 .

2.7 Mutation Process and New Population Generation

The procedure of mutations begins by randomly selecting a number of solutions (widows) from the pop_1 which will be mutated individually. Two cells from each selected solution are randomly exchanged, and the new mutation solutions will be kept in pop_3 . The new generation can finally be generated as a combination of pop_2 and pop_3 , which will then be evaluated to return the optimal solution (W^*) of values bearing the N dimension.

In the BBWO-HCA, the cannibalism rate (C_R), the procreation rate (P_r), and the mutation rate (M_r) are used as parameters. The value of the (C_R) is determined by the fitness values obtained by Eq. 1, and the P_r and M_r are identical to those of the standard BWO.

2.8 HCA Steps

The algorithm uses the best solution (W^*) of the BBWO as an initial solution for the HCA. The solution is modified by selecting one feature randomly and flipping the value of that feature, i.e., if the feature value is "0" it is changed to "1" (which indicates adding one feature), and if the value is "1" it is changed to "0" (which indicates deleting one feature). If the fitness value of the modified

solution is improved, it will replace the old one, otherwise, it discards the new solution.

Next, the HCA iteration counter and BBWO best solution (W^*) are updated, and the stopping criteria of the BBWO is checked. If the BBWO stopping condition is met, i.e., the max iterations are reached, the algorithm stops and returns the best solution (W^*), otherwise, a new iteration for the BBWO starts.

The pseudocode for the BBWO is shown in Fig. 1 and the additional steps involved in implementing the HCA are shown in Fig. 2. Together, they form the pseudocode for the BBWO-HCA.

3 Experiment Setup and Results

28 well-known datasets from the University of California Irvine (UCI)¹ machine learning repository have been used to investigate the performance and strength of our proposed methods. The dataset is randomly split into 80% for the training set and 20% for the test set. These rates are widely accepted data partition rates. The datasets vary in the number of features and instances. Table 1 presents a brief description of the datasets. Each row in the table represents the number of features, objects, classes, and the domain to which each of these datasets belong.

The performance of our proposed method, the BBWO-HCA, is compared with six well-respected binary FS algorithms: (BPSO [10], BMVO [2], BGWO [1, 7], BMFO [19], BWOA [11], BBAT [13]) based on the two evaluation criteria, classification *accuracy* and the number of *features selected*.

To ensure an impartial comparison and a correct evaluation between our proposed method and other FS algorithms, we re-implemented the six FS algorithms using the same parameters values as illustrated in Table 2 and the same transformation function as explained in Sect. 2.4. The algorithms are run independently multiple times and the average accuracy and the average number of features selected are reported.

3.1 BBWO-HCA vs. BBWO Results and Discussion 1

Table 3 shows the comparison between our two algorithms (BBWO-HCA and BBWO algorithms) based on the two evaluation criteria (the classification accuracy and feature selected). The best *classification accuracy* and the *lower number of features selected* are highlighted in bold.

The results show that the BBWO-HCA is more efficient than the BBWO in terms of maximizing classification accuracy. The BBWO-HCA outperforms the BBWO in 15 datasets and obtains the same results in 13 datasets in terms of classification accuracy. When considering the average accuracy for all datasets, the performance of the BBWO-HCA is better than the BBWO. This is shown in Fig. 3.

¹ The datasets can be downloaded here: <https://archive.ics.uci.edu/ml/datasets.php>.

```

1  # Set the value of the parameters
2  population size ( $N_{pop}$ ) = 20; number of iterations (maxIteration) = 10; number of
3  features ( $N_f$ ) = dimension size; procreate rate ( $P_r$ ) = 0.6; mutation rate ( $M_r$ ) = 0.4
4  # Initialization process
5  Generating the initial population of solutions randomly ( $N_{pop} \times N_f$ ).
6  Each solution represents one widow, which is indicated in one-dimension vector  $1 \times N_f$ 
7  Calculate the fitness value for each solution using Eq.1
8  Evaluate all solutions in the population based on their fitness value and save them in
9   $pop_1$ 
10 Set the best solution in the population as  $W^*$ 
11 based on  $P_r$  calculate the number of reproductions  $N_r$ 
12 based on  $M_r$  calculate the number of mutations  $N_m$ 
13 Define  $I=0$ 
14 while  $I < \text{maxIteration}$  do
15     # and cannibalism processes
16     for  $i = 1$  to ( $N_r / 2$ ) do
17         Randomly select two solutions  $w_1, w_2$  as parents from  $pop_1$ 
18         Generate two children  $c_1, c_2$  using Eq. 4
19         Transformation  $c_1, c_2$  to binary nature using Eq.2 and 3
20         Calculate the fitness value of  $c_1, c_2$  using Eq.1
21         Destroy the father  $w_1$  or  $w_2$  based on their fitness value (cannibalism process)
22          $c_1$  or  $c_2$  based on their fitness value (sibling cannibalism)
23         Save the remaining solutions in  $pop_2$ 
24     end for
25     #Mutation process
26     for  $i = 1$  to  $N_m$  do
27         Randomly select a solution from  $pop_1$ 
28         Apply the mutation process on the selected solution
29         Save the result (the new solution) in  $pop_3$ 
30     end for
31     #Update the population
32     Update the population =  $pop_2 + pop_3$ 
33     Evaluate all solutions in the population using Eq.1
34     Update  $W^*$  if there is a better solution
35      $I = I + 1$ 
36 end while
37 returning the best solution  $W^*$ 

```

Fig. 1. The Binary Black Widow Algorithm Pseudocode for FS

In terms of minimizing the total number of features selected, the results show that the BBWO-HCA obtains better results than the BBWO in 26 datasets, the same results in one dataset, and worse results in one dataset. The results also show that on average, the BBWO-HCA is more efficient than the BBWO in this regard. This is shown in Fig. 4.

```

38 #Hill climbing algorithm
39 #Set the new solution = the best solution
40  $S^* = W^*$ 
41 #Set the fitness value of  $S^*$  = fitness values of  $W^*$ 
42  $F(S^*) = F(W^*)$ 
43  $H = 1$ 
44 While  $H < \text{maxIteration}$  do
45    $S = S^*$ 
46   Randomly select one feature (i) in  $S^*$ ,  $i = 1, 2, \dots, N_f$ 
47   if  $S_i^* = 0$ , then  $S_i^* = 1$ ; else  $S_i^* = 0$ 
48   if  $F(S^*) < F(W^*)$  then  $S^* = W^*$  else  $S^* = S$ 
49    $H = H + 1$ 
50 end while
51  $I = I + 1$ 
52 end while
53 returning the best solution =  $W^*$ 
54

```

Fig. 2. BBWO with the Hill-climbing (BBWO-HCA) Algorithm**Table 1.** Datasets description.

No.	Datasets	Features	Objects	Classes	Domain
1	Breastcancer	9	699	2	Medical
2	BreastEW	30	569	2	Medical
3	CongressEW	16	435	2	Politics
4	Exactly	13	1000	2	Medical
5	Exactly2	13	1000	2	Medical
6	HeartEW	13	270	5	Medical
7	IonosphereEW	34	351	2	Electronic
8	Lymphography	18	148	4	Medical
9	M-of-n	13	1000	2	Medical
10	PenglungEW	325	73	2	Medical
11	SonarEW	60	208	2	Medical
12	SpectEW	22	267	2	Medical
13	Tic-tac-toe	9	958	2	Game
14	Vote	16	300	2	Politics
15	WaveformEW	40	5000	3	Physical
16	Zoo	16	101	7	Artificial
17	Colon	2000	62	2	Medical
18	Parkinsons	22	195	2	Medical
19	Lungcancer	21	226	2	Medical
20	Leukemia	7129	72	2	Medical
21	Dermatology	34	366	6	Medical
22	Semeion	256	1593	10	Handwriting
23	Satellite	36	5100	2	Physical
24	Spambase	57	4601	2	Computer
25	Segment	19	2310	7	Images
26	Credit	20	1000	2	Business
27	KrvskpEW	36	3196	2	Game
28	Plants-100	64	1599	100	Agriculture

Table 2. BBWO-HCA Parameters

Parameter Name	Value	Parameter Name	Value
Population-size	20	No. of iterations	10
Number of independent runs	20	K (KNN classifier)	5
Dimension-size	No. of features	Number of iterations for hill climbing	20
pr(procreate rate)	0.6	mr (mutation rate)	0.4
α	0.99	β	0.01

Table 3. Compression between BBWO-HCA and BBWO

Datasets	Classification accuracy		Feature selected	
	BBWO-HCA	BBWO	BBWO-HCA	BBWO
Breastcancer	0.98	0.97	3.00	3.00
BreastEW	0.95	0.94	4.60	12.25
CongressEW	0.95	0.95	1.50	4.60
Exactly	1	0.91	5.25	3.75
Exactly2	0.77	0.77	2.00	3.65
HeartEW	0.85	0.84	2.55	3.80
IonosphereEW	0.90	0.88	9.45	13.75
Lymphography	0.85	0.85	4.05	6.80
M-of-n	1	0.95	5.75	7.00
PenglungEW	0.90	0.90	100.85	151.75
SonarEW	0.87	0.86	14.75	24.40
SpectEW	0.82	0.81	6.50	8.50
Tic-tac-toe	0.82	0.80	4.05	3.80
Vote	0.95	0.93	1.55	4.05
WaveformEW	0.88	0.88	19.80	20.60
Zoo	0.92	0.92	4.60	5.05
Parkinsons	0.90	0.90	2.65	7.70
Lungcancer	0.92	0.90	4.70	7.00
Colon	0.89	0.87	888.35	980.22
Leukemia	0.86	0.86	3499.75	3531.90
Dermatology	0.97	0.97	11.25	14.70
Semeion	0.94	0.93	120.00	130.00
Satellite	0.99	0.99	4.40	9.20
Spambase	0.93	0.93	21.10	28.60
Segment	0.96	0.96	6.60	7.70
Credit	0.79	0.79	6.00	7.22
KrvskpEW	0.97	0.95	15.80	19.00
Plants-100	0.81	0.80	32.20	32.50
Average	0.9050	0.8932	171.53	180.44
Rank	1	2	1	2

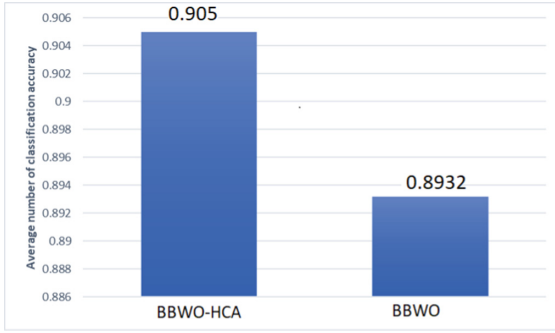


Fig. 3. Average of classification accuracy of BBWO-HCA vs. BBWO

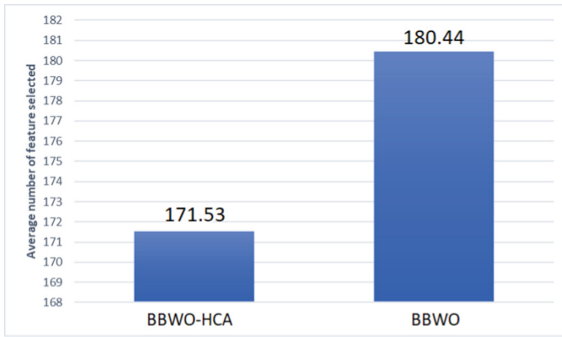


Fig. 4. Average of feature selection of BBWO-HCA vs. BBWO

3.2 BBWO-HCA vs. Six FS Algorithms, Results and Discussion 2

We compared the BBWO-HCA results with six FS algorithms (BPSO, BMVO, BGWO, BMFO, BWOA, BBAT). The results are presented in Tables 4 and 5. The test results reveal that the **BBWO-HCA** outperforms all six algorithms unless an algorithm already reached the best possible solution, in which case the BBWO-HCA results are the same as the other algorithm. For example, the **BBWO-HCA** produced better results than the **BPSO** in 26 datasets and the same results in two datasets.

In to the number of features selected, the test results reveal that the BBWO-HCA outperforms all six FS algorithms for all 28 datasets tested, see Table 5. These results are depicted pictorially in Figs. 5 and 6, and they show that the BBWO-HCA is an effective algorithm for solving FS problems.

Table 4. Comparison BBWO-HCA with all algorithms based on the classification accuracy

Datasets Name	BBWO HCA	BPSO	BMVO	BGWO	BMFO	BWOA	BBAT
Breastcancer	0.98	0.96	0.97	0.96	0.97	0.97	0.96
BreastEW	0.95	0.94	0.94	0.95	0.94	0.93	0.94
CongressEW	0.95	0.92	0.95	0.95	0.95	0.95	0.94
Exactly	1	0.76	0.89	0.74	0.90	0.91	0.73
Exactly2	0.77	0.77	0.76	0.75	0.76	0.74	0.74
HeartEW	0.85	0.81	0.85	0.84	0.85	0.85	0.82
IonosphereEW	0.90	0.86	0.88	0.88	0.88	0.88	0.88
Lymphography	0.85	0.82	0.84	0.82	0.85	0.83	0.81
M-of-n	1	0.83	0.99	0.88	0.98	0.98	0.81
PenglungEW	0.90	0.87	0.89	0.89	0.89	0.88	0.88
SonarEW	0.87	0.86	0.87	0.86	0.86	0.87	0.86
SpectEW	0.82	0.81	0.81	0.82	0.82	0.81	0.81
Tic-tac-toe	0.82	0.74	0.81	0.78	0.82	0.81	0.76
Vote	0.95	0.91	0.94	0.94	0.94	0.94	0.93
WaveformEW	0.88	0.86	0.88	0.87	0.88	0.88	0.83
Zoo	0.92	0.89	0.89	0.88	0.88	0.90	0.89
Parkinsons	0.90	0.88	0.89	0.86	0.89	0.89	0.88
Lungcancer	0.92	0.88	0.91	0.90	0.91	0.91	0.90
Colon	0.89	0.86	0.89	0.87	0.87	0.87	0.87
Leukemia	0.86	0.83	0.86	0.85	0.86	0.86	0.85
Dermatology	0.97	0.89	0.96	0.95	0.97	0.97	0.92
Semeion	0.94	0.92	0.93	0.92	0.93	0.93	0.92
Satellite	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Spambase	0.93	0.88	0.93	0.92	0.93	0.93	0.89
Segment	0.96	0.94	0.96	0.96	0.96	0.96	0.94
Credit	0.79	0.76	0.79	0.77	0.78	0.79	0.78
KrvskpEW	0.97	0.90	0.96	0.95	0.97	0.97	0.87
Plants-100	0.81	0.78	0.79	0.78	0.80	0.79	0.77
Average	0.9050	0.8614	0.8935	0.8760	0.8939	0.8925	0.8632
Rank	1	7	3	5	2	4	6

Table 5. Comparison BBWO-HCA all algorithms based on the features selected

Datasets	BBWO-HCA	BPSO	BMVO	BGWO	BMFO	BWOA	BBAT
Breastcancer	3.00	3.40	4.55	5.15	4.35	4.60	3.45
BreastEW	4.60	11.40	10.95	13.55	13.20	12.12	13.15
CongressEW	1.50	5.20	4.25	5.95	5.40	4.20	5.55
Exactly	5.25	5.30	7.25	7.05	7.05	6.65	5.80
Exactly2	2.00	3.95	2.33	5.40	3.15	2.10	3.50
HeartEW	2.55	4.25	3.45	4.15	3.70	3.45	4.65
IonosphereEW	9.45	14.35	13.35	15.90	15.00	12.55	16.55
Lymphography	4.05	7.60	6.66	7.56	7.35	6.35	7.55
M-of-n	5.75	5.50	7.22	8.00	6.75	7.35	6.15
PenglungEW	100.85	154.80	152.35	155.20	152.45	146.35	156.85
SonarEW	14.75	27.05	25.55	28.05	28.95	23.85	27.00
SpectEW	6.50	8.95	8.22	10.15	8.20	8.75	9.85
Tic-tac-toe	4.05	4.20	4.55	4.55	4.41	4.05	4.28
Vote	1.55	5.05	4.95	6.35	5.85	4.50	6.15
WaveformEW	19.80	22.00	22.15	21.45	21.35	19.45	20.25
Zoo	4.60	5.59	6.35	6.65	6.13	5.75	6.50
Parkinsons	2.65	8.00	8.45	9.15	9.10	8.20	9.25
Lungcancer	4.70	7.25	8.66	9.35	8.90	8.05	8.95
Colon	888.35	961.65	963.25	965.55	962.15	943.55	963.35
Leukemia	3499.75	3555.82	3571.85	3535.85	3534.55	3511.35	3513.50
Dermatology	11.25	16.85	15.95	16.70	16.60	16.45	16.50
Semeion	120.00	131.85	127.00	128.6	131.60	126.80	126.70
Satellite	4.40	13.01	10.55	12.40	11.40	10.10	12.45
Spambase	21.10	29.77	26.55	30.50	26.25	26.50	27.25
Segment	6.60	8.72	9.25	9.90	9.95	8.90	9.60
Credit	6.00	8.41	7.95	8.50	8.30	7.72	8.75
KrvskpEW	15.80	19.73	19.81	21.30	18.56	17.92	18.45
Plants-100	32.20	35.55	33.15	33.80	33.32	34.15	35.50
Average	171.53	181.61	181.66	181.66	180.85	178.27	180.26
Rank	1	5	6	6	4	2	3

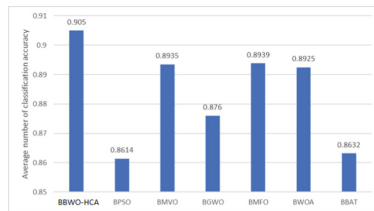


Fig. 5. Average number of classification accuracy of all algorithms

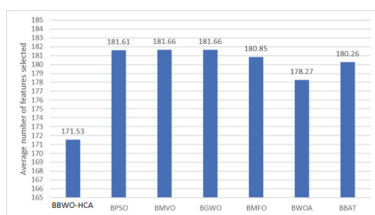


Fig. 6. Average number of features selected of all algorithms

4 Conclusion and Future Works

Recently, a novel algorithm, the Black Widow Algorithm (BWO), has been developed to solve optimization problems. BWO is derived from nature; it mimics the singular mating behaviour exhibited by the black widow spider.

Initially, we developed the BBWO algorithm based on the BWO for solving FS problems. In this work, we further improved the BBWO by combining it with the Hill-Climbing Algorithm. The newly developed algorithm, BBWO-HCA, is tested using 28 UCI datasets and compared with six well-regarded algorithms in the domain. The test results show that the BBWO-HCA outperforms the BBWO and almost all comparable algorithms for most datasets tested.

This work opened the door for further FS and optimization studies. Examples of such studies are:

- The test results of the BBWO-HCA can be further analyzed to determine the impact of the dataset size, number of features in the dataset, number of instances, etc. on the performance of the algorithm.
- Combining the BBWO with other algorithms and studying the outcomes of these new combinations are open future works.
- The BBWO and BBWO-HCA can be applied to various other areas of study to solve many other real-world optimization problems such as text mining, clustering, image processing, and routing problems.

References

1. Abdel-Basset, M., El-Shahat, D., El-henawy, I., de Albuquerque, V.H.C., Mirjalili, S.: A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Syst. Appl.* **139**, 112824 (2020)
2. Al-Madi, N., Faris, H., Mirjalili, S.: Binary multi-verse optimization algorithm for global optimization and discrete problems. *Int. J. Mach. Learn. Cybern.* **10**(12), 3445–3465 (2019). <https://doi.org/10.1007/s13042-019-00931-8>
3. Al-Saedi, A., Mawlood-Yunis, A.R.: Binary black widow optimization algorithm for feature selection problems. In: Simos, D.E., Rasskazova, V.A., Archetti, F., Kotsireas, I.S., Pardalos, P.M. (eds.) *Learning and Intelligent Optimization (LION 2022)*. LNCS, vol. 13621, pp. 93–107. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-24866-5_7

4. Alweshah, M., Alkhalaileh, S., Albashish, D., Mafarja, M., Bsoul, Q., Dorgham, O.: A hybrid mine blast algorithm for feature selection problems. *Comput.* **25**(1), 517–534 (2020). <https://doi.org/10.1007/s00500-020-05164-4>
5. Arora, S., Anand, P.: Binary butterfly optimization approaches for feature selection. *Expert Syst. Appl.* **116**, 147–160 (2019)
6. Dordaie, N., Navimipour, N.J.: A hybrid particle swarm optimization and hill climbing algorithm for task scheduling in the cloud environments. *ICT Express* **4**(4), 199–202 (2018)
7. Emary, E., Zawbaa, H.M., Hassanien, A.E.: Binary grey wolf optimization approaches for feature selection. *Neurocomputing* **172**, 371–381 (2016)
8. Hammouri, A.I., Mafarja, M., Al-Betar, M.A., Awadallah, M.A., Abu-Doush, I.: An improved dragonfly algorithm for feature selection. *Knowl.-Based Syst.* **203**, 106131 (2020)
9. Hayyolalam, V., Kazem, A.A.P.: Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **87**, 103249 (2020)
10. Mafarja, M., Jarrar, R., Ahmad, S., Abusnaina, A.A.: Feature selection using binary particle swarm optimization with time varying inertia weight strategies. In: *Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*, pp. 1–9 (2018)
11. Mafarja, M., Mirjalili, S.: Whale optimization approaches for wrapper feature selection. *Appl. Soft Comput.* **62**, 441–453 (2018)
12. Mirjalili, S.: The ant lion optimizer. *Adv. Eng. Softw.* **83**, 80–98 (2015)
13. Mirjalili, S., Mirjalili, S.M., Yang, X.S.: Binary bat algorithm. *Neural Comput. Appl.* **25**(3), 663–681 (2014)
14. Mostafa, R.R., Ewees, A.A., Ghoniem, R.M., Abualigah, L., Hashim, F.A.: Boosting chameleon swarm algorithm with consumption AEO operator for global optimization and feature selection. *Knowl.-Based Syst.* **246**, 108743 (2022)
15. Neggaz, N., Houssein, E.H., Hussain, K.: An efficient henry gas solubility optimization for feature selection. *Expert Syst. Appl.* **152**, 113364 (2020)
16. Syriopoulos, P.K., Kotsiantis, S.B., Vrahatis, M.N.: Survey on KNN methods in data science. In: Simos, D.E., Rasskazova, V.A., Archetti, F., Kotsireas, I.S., Pardalos, P.M. (eds.) *Learning and Intelligent Optimization (LION 2022)*. LNCS, vol. 13621, pp. 379–393. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-24866-5_28
17. Venkatesh, B., Anuradha, J.: A review of feature selection and its methods. *Cybern. Inf. Technol.* **19**(1), 3–26 (2020)
18. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N. (eds.) *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. *Studies in Computational Intelligence*, vol. 284, pp. 65–74. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12538-6_6
19. Zawbaa, H.M., Emary, E., Parv, B., Sharawi, M.: Feature selection approach based on moth-flame optimization algorithm. In: *2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 4612–4617 (2016)