

Binary Dragonfly Algorithm for Feature Selection

Majdi M. Mafarja, Derar Eleyan, Iyad Jaber

Department of Computer Science
Birzeit University
Birzeit, West Bank, Palestine

Seyedali Mirjalili

School of Information and Communication
Technology
Griffith University, Nathan Campus
Brisbane, QLD 4111, Australia

Abdelaziz Hammouri

Department of Computer Information Systems
Al-Balqa' Applied University, Amman, Jordan
aziz@bau.edu.jo

Abstract: Wrapper feature selection methods aim to reduce the number of features from the original feature set to and improve the classification accuracy simultaneously. In this paper, a wrapper-feature selection algorithm based on the binary dragonfly algorithm is proposed. Dragonfly algorithm is a recent swarm intelligence algorithm that mimics the behavior of the dragonflies. Eighteen UCI datasets are used to evaluate the performance of the proposed approach. The results of the proposed method are compared with those of Particle Swarm Optimization (PSO), Genetic Algorithms (GAs) in terms of classification accuracy and number of selected attributes. The results show the ability of Binary Dragonfly Algorithm (BDA) in searching the feature space and selecting the most informative features for classification tasks.

Keywords—Feature Selection; Optimization; Dragonfly Algorithm, classification.

I. INTRODUCTION:

Feature selection (FS) is a pre-processing technique that has been proven to significantly impact the performance of the data mining techniques (*e.g.* classification) [1] in terms either the quality of the extracted patterns or the running time required to analyze the complete dataset. FS aims to improve the classification accuracy by eliminating the redundant, irrelevant and noisy data from the original dataset. FS methods are categorized (based on the evaluation criteria of the selected subset) into filters and wrappers [2]. While the wrapper approaches concedes a learning algorithm (*e.g.* classification) in the evaluation of the feature subset, filters depend on the data itself to evaluate the feature subset using designated methods (*e.g.* information gain) [3].

Searching an optimal subset of features is a big challenge when solving feature selection problems. The main objective when selecting feature is to find a set of M features from an original set of N where $M < N$ without information loose. Therefore, generating

all possible subsets is an impractical solution to this problem. If the dataset includes N features, then there will be 2^N subsets to be generated and evaluated, which is considered as a computationally expensive task [4].

In literature, many feature selection methods adopted metaheuristic algorithms to search for the optimal subset. An Ant Lion Optimizer (ALO) [5] that mimics the behavior of antlions in hunting prey is employed as a wrapper FS method in [6]. Grey Wolf Optimizer (GWO) is a recent algorithm [7] that has been successfully employed for solving feature selection problems in [8, 9]. Another recent wrapper was proposed in [10], in which the main algorithm is Moth-flame Optimization algorithm [11]

Genetic Algorithm (GA) [12] is an evolution-based algorithm that shows a good performance in solving non-linear and complex problems [13]. Different genetic algorithm approaches have been proposed to tackle the feature selection [13-17]. Another feature selection approach that based on Scatter Search (SSAR) is proposed by Jue *et al.* in [18]. The Particle Swarm Optimization algorithm (PSO), proposed by Kennedy and Eberhart [19], was used in feature selection approaches as in [20-23]. Based on the biological behaviour of bees, Karaboga [24] proposed an optimisation approach called the Artificial Bee Colony (ABC) algorithm. A feature selection for intrusion detection systems using binary ABC is proposed in [25].

The main objective of this paper is to propose a wrapper feature selection approach that uses the recent BDA algorithm as a search strategy and the K-Nearest Neighborhood (*KNN*) classifier as an evaluator. The rest of the paper is organized as

follows: section 2 describes the BDA algorithm. In section 3, the details of the proposed approach are discussed. The experimental results are presented in section 4 while the conclusions and future are outlined in section 5.

II. DRAGONFLY ALGORITHM

As the name shows, the DA algorithm has been inspired from dragonflies [26]. This algorithm can be considered as a swarm intelligence technique to estimate the global optimum of a given optimization problem. The swarming behavior of dragonflies and the mathematical models to implement it are given below: [27]:

- Separation refers to the mechanism that individuals follow to avoid collision with other neighbor individuals. This behavior is mathematically modelled as in Eq. (1).

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

where X represents the current individual's position, X_j represents the j -th neighboring individual of the X position, and N is the neighborhood size.

- Alignment indicates the individuals velocity matching according to other near individuals. This behavior is mathematically modelled as in Eq. (2).

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

- where V_j represents the j -th neighborhood individual's velocity and N is the neighborhood size.
- Cohesion refers to the individuals' tendency the neighborhood's center of mass. This behavior is mathematically modelled as in Eq. (3).

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

where X represents the current individual's position, X_j represents the j -th neighboring individual of the X position, and N is the neighborhood size.

Attraction towards the food source and escaping from enemies are other two key behaviors that each individual behaves to survive. The attraction towards the food is modelled as in Eq. (4).

$$F_i = X^+ - X \quad (4)$$

where X^+ represents the position of the food source and X represents the current individual's position.

$$E_i = X^- + X \quad (5)$$

where X^- represents the enemy's position and X represents the current individual's position.

DA utilized two vectors to solve optimization problems: step vector and position vector. These two vectors are defined. The step vector is defined as follows:

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

where s represents the separation weight, S_i shows the separation of the i -th individual, a is the alignment weight, A_i is the alignment of i -th individual, c indicates the cohesion weight, C_i is the cohesion of the i -th individual, f is the food factor, F_i is the food source of the i -th individual, e is the enemy factor, E_i is the position of enemy of the i -th individual, w is the inertia weight, and t is the iteration number.

In a continuous search space, the position of dragonflies is updated by adding the step vector to the previous position. In a binary search space, however, the following equations should be used:

$$X_{t+1} = \begin{cases} -X_t, & x < T(\Delta x_{t+1}) \\ X_t, & x \geq T(\Delta x_{t+1}) \end{cases} \quad (7)$$

where r is a random number in the range $[0, 1]$, $T(\Delta x_{t+1})$ is calculated as in Eq. (8).

$$T(\Delta x) = \left\lfloor \frac{\Delta x}{\sqrt{\Delta x^2 + 1}} \right\rfloor \quad (8)$$

The pseudocode of BDA algorithm is given in Algorithm 1.

Initialize the population X_i ($i = 1, 2, \dots, n$)

Initialize ΔX_i ($i = 1, 2, \dots, n$)

while ($t < \text{Max_Iteration}$)

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 using Eq. (6)

Calculate $T(\Delta x)$ using Eq. (8)

Update X_{t+1} using Eq. (7)

end while

Algorithm 1. Pseudo-code of the BDA algorithm

III. THE PROPOSED BDA ALGORITHM

FS is considered as a binary optimization problem, where solutions are restricted to the binary $\{0, 1\}$ values. Therefore, the binary version of the DA algorithm is suitable to be used to solve this problem. In this work, a vector of *zeros* and *ones* is used to represent the solution of the problem, where the zero element indicates that the corresponding feature does not selected and the one element mean that this feature is selected. The length of the solution vector is the number of features in the original dataset.

In this work, a wrapper feature selection method that based on the DA algorithm is proposed. The *KNN* classifier is used to evaluate the selected feature subsets. FS is considered as a multi-objective problem, where two contradictory objectives are to be achieved; to maximize the classification accuracy while trying to minimize the number of selected features. Therefore, the cardinality of the reduct is considered in the objective function besides the classification error rate. Eq. (9) represents the objective function.

$$Fitness = \alpha \gamma_R(D) + \beta \frac{|R|}{|C|} \quad (9)$$

where $\gamma_R(D)$ represents the classification error rate of a given classier (the *K*-nearest neighbor (*KNN*) classifier is used here). Furthermore, $|R|$ is the cardinality of the selected subset and $|C|$ is the total number of features in the dataset, α and β are two parameters corresponding to the importance of classification quality and subset length, $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$ adopted from [6].

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. Data sets and Parameters

Table I shows the details of the 18 datasets that we used to assess the performance of the proposed algorithm. They are well known benchmark datasets taken from the UCI data repository [28]. A wrapper approach-based on the *KNN* classifier (where $K = 5$ [6]) with the Euclidean distance metric is used to generate the best reduct.

In the proposed approach, each dataset is divided into three parts:

- The first part is used to train the classifier through optimization and at the final evaluation.
- The second part is for assessing the performance of the classifier at the optimization time.
- The third part is to evaluate the best selected features given the trained classifier [10]. □

In K -fold cross-validation, $K - 1$ folds are used for training and validation and the remaining fold is used for testing. This process is repeated M times. Hence, individual optimizer is evaluated $K * M$ times for each data set. The training and validation data are equally sized. In all experiments, the parameters are set as follows: The maximum number of iterations is 100 and the population size is 10. Furthermore, each algorithm is run 20 times with random seed on an Intel Core i5 machine, 2.2 GHz CPU and 4 GB of RAM. Please note that the parameters of SA are identical to those used in the previous subsection.

TABLE I. LIST OF DATASETS USED IN THE EXPERIMENTS

	Dataset	No. of Attributes	No. of Objects
1.	Breastcancer	9	699
2.	BreastEW	30	569
3.	CongressEW	16	435
4.	Exactly	13	1000
5.	Exactly2	13	1000
6.	HeartEW	13	270
7.	IonosphereEW	34	351
8.	KrvskpEW	36	3196
9.	Lymphography	18	148
10.	M-of-n	13	1000
11.	PenglungEW	325	73
12.	SonarEW	60	208
13.	SpectEW	22	267
14.	Tic-tac-toe	9	958
15.	Vote	16	300
16.	WaveformEW	40	5000
17.	WineEW	13	178
18.	Zoo	16	101

To assess the performance of the proposed algorithm, two state-of-the-art FS methods (PSO and GA) are used for comparison purposes, and all algorithms are compared to the performance gained by using all features. The parameter setting for the these different optimization algorithms is obtained from [10]. Different evaluation criteria are used as follows:

- The mean classification accuracy obtained from the 20 runs. From each run, the accuracy of the best solution is considered.
- The average selection size from the 20 runs. In each run, the cardinality of the best solution is considered.
- Different statistics (mean, min and max) for fitness values obtained from each approach are reported.

- Statistical standard deviation (*std*) is reported compare for all approaches to indicate the stability and robustness of the optimizer.

B. Results and discussion

Table II shows the average classification accuracy gained when using all features in the classification process, besides the results obtained by using the selected features by the different FS algorithms. We can remark that the wrapper-feature selection approaches, in general, proved their capability in the FS problem. Inspecting the results in Table II, it is evident that BDA is better than other approaches since it obtained better results in most of the datasets.

In Table III, the kept feature ratio to the total number of features using BDA and other approaches are reported. BDA shows much better performance than other approaches on most of the datasets.

The good performance of BDA algorithm proves its ability to adaptively search the feature space for the most informative features and avoid premature convergence that may be caused by falling in local minima. In addition to proving its capability to balance between the exploration and exploitation during optimization.

TABLE II. AVERAGE CLASSIFICATION ACCURACY

Data set	All	GA	PSO	DF
Breastcancer	0.961	0.974	0.970	0.963
BreastEW	0.945	0.967	0.964	0.961
CongressEW	0.921	0.967	0.964	0.967
Exactly	0.662	0.717	0.714	0.980
Exactly2	0.738	0.742	0.749	0.745
HeartEW	0.782	0.853	0.829	0.830
IonosphereEW	0.812	0.853	0.862	0.930
KrvskpEW	0.907	0.953	0.953	0.953
Lymphography	0.712	0.848	0.820	0.877
M-of-n	0.856	0.892	0.927	0.992
PenglungEW	0.715	0.794	0.756	0.895
SonarEW	0.671	0.843	0.831	0.915
SpectEW	0.796	0.865	0.856	0.853
Tic-tac-toe	0.724	0.765	0.769	0.788
Vote	0.908	0.960	0.960	0.958
WaveformEW	0.759	0.782	0.784	0.750
WineEW	0.940	0.980	0.977	0.980
Zoo	0.811	0.897	0.901	0.958

TABLE III. AVERAGE SELECTION RATION

Data set	GA	PSO	DF
Breastcancer	0.511	0.511	0.550
BreastEW	0.460	0.440	0.395
CongressEW	0.362	0.400	0.288
Exactly	0.523	0.492	0.469
Exactly2	0.215	0.462	0.208
HeartEW	0.508	0.538	0.527
IonosphereEW	0.418	0.424	0.338
KrvskpEW	0.539	0.533	0.493
Lymphography	0.400	0.400	0.453
M-of-n	0.677	0.600	0.465
PenglungEW	0.434	0.387	0.380
SonarEW	0.420	0.450	0.458
SpectEW	0.373	0.464	0.361
Tic-tac-toe	0.556	0.556	0.661
Vote	0.362	0.512	0.259
WaveformEW	0.580	0.520	0.524
WineEW	0.477	0.523	0.485
Zoo	0.362	0.400	0.356

In Tables IV, V and VI, simple statistical analysis is conducted to assess the performance of the proposed algorithm on average. First, it is obvious that the fitness values achieved obtained when using all features are the worst ever. It is also apparent from the tables that BDA has competitive results compared with other methods.

TABLE IV. THE WORST (MAX) FITNESS VALUE

Data set	All	GA	PSO	DF
Breastcancer	0.060	0.030	0.039	0.049
BreastEW	0.068	0.047	0.058	0.049
CongressEW	0.097	0.048	0.062	0.048
Exactly	0.362	0.312	0.342	0.023
Exactly2	0.266	0.270	0.270	0.286
HeartEW	0.200	0.178	0.189	0.195
IonosphereEW	0.188	0.171	0.162	0.094
KrvskpEW	0.110	0.061	0.065	0.038
Lymphography	0.367	0.180	0.220	0.167
M-of-n	0.192	0.156	0.117	0.005
PenglungEW	0.417	0.320	0.320	0.165
SonarEW	0.406	0.214	0.229	0.109
SpectEW	0.270	0.180	0.191	0.197
Tic-tac-toe	0.300	0.260	0.260	0.238
Vote	0.130	0.050	0.080	0.069
WaveformEW	0.249	0.244	0.224	0.262
WineEW	0.085	0.033	0.033	0.038
Zoo	0.353	0.152	0.152	0.139

To show how stable the algorithms are, the standard deviations are presented in Table VII. It may be seen that the BDA algorithm is highly reliable. And robust

TABLE V. THE BEST (MIN) FITNESS VALUE

Data set	All	GA	PSO	DF
Breastcancer	0.030	0.017	0.022	0.030
BreastEW	0.058	0.021	0.026	0.025
CongressEW	0.048	0.021	0.014	0.021
Exactly	0.308	0.258	0.189	0.005
Exactly2	0.228	0.234	0.231	0.226
HeartEW	0.144	0.089	0.133	0.145
IonosphereEW	0.137	0.120	0.111	0.042
KrvskpEW	0.067	0.032	0.036	0.022
Lymphography	0.204	0.140	0.140	0.072
M-of-n	0.132	0.084	0.018	0.005
PenglungEW	0.042	0.120	0.160	0.031
SonarEW	0.261	0.100	0.086	0.034
SpectEW	0.180	0.079	0.101	0.107
Tic-tac-toe	0.231	0.204	0.204	0.182
Vote	0.060	0.030	0.010	0.018
WaveformEW	0.230	0.202	0.202	0.228
WineEW	0.034	0.017	0.017	0.005
Zoo	0.125	0.063	0.069	0.001

TABLE VI. THE AVERAGE FITNESS VALUE

Data set	All	GA	PSO	DF
Breastcancer	0.040	0.026	0.030	0.038
BreastEW	0.061	0.033	0.036	0.040
CongressEW	0.074	0.033	0.036	0.035
Exactly	0.332	0.283	0.286	0.006
Exactly2	0.249	0.258	0.251	0.249
HeartEW	0.176	0.147	0.171	0.167
IonosphereEW	0.157	0.147	0.138	0.071
KrvskpEW	0.087	0.047	0.047	0.031
Lymphography	0.269	0.152	0.180	0.125
M-of-n	0.169	0.108	0.073	0.005
PenglungEW	0.285	0.206	0.244	0.089
SonarEW	0.333	0.157	0.169	0.066
SpectEW	0.220	0.135	0.144	0.144
Tic-tac-toe	0.264	0.235	0.231	0.209
Vote	0.094	0.040	0.040	0.041
WaveformEW	0.238	0.218	0.216	0.244
WineEW	0.061	0.020	0.023	0.020
Zoo	0.219	0.103	0.099	0.039

TABLE VII. THE STANDARD DEVIATION

Data set	All	GA	PSO	DF
Breastcancer	0.013	0.005	0.007	0.014
BreastEW	0.005	0.010	0.013	0.014
CongressEW	0.021	0.014	0.020	0.008
Exactly	0.021	0.021	0.058	0.077
Exactly2	0.018	0.015	0.018	0.033
HeartEW	0.021	0.040	0.022	0.026
IonosphereEW	0.020	0.020	0.023	0.017
KrvskpEW	0.016	0.012	0.011	0.047
Lymphography	0.074	0.018	0.028	0.026
M-of-n	0.027	0.029	0.046	0.038
PenglungEW	0.159	0.088	0.075	0.046
SonarEW	0.063	0.057	0.060	0.029
SpectEW	0.041	0.038	0.036	0.035
Tic-tac-toe	0.032	0.022	0.022	0.039
Vote	0.029	0.010	0.027	0.018
WaveformEW	0.007	0.016	0.008	0.017
WineEW	0.019	0.007	0.009	0.019
Zoo	0.085	0.034	0.031	0.058

V. CONCLUSION

This paper proposed a new wrapper feature selection using the dragonfly algorithm. The main goal of the proposed FS approach was to identify the minimal reduct that could obtain a higher accuracy than using all features in the dataset. A set of well-known FS datasets from UCI data repository are used to evaluate the proposed approach, and the results are compared with the GA and PSO algorithms. The experimental results show a superior performance for the BDA approach compared to other approaches.

As a future work, it is worth to employ different transfer functions and see how BDA behaves accordingly.

REFERENCES

- [1] S. F. Crone, S. Lessmann, and R. Stahlbock, "The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing," *European Journal of Operational Research*, vol. 173, pp. 781-800, 2006.
- [2] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*. Boston: Kluwer Academic Publishers 1998.
- [3] Z. Zhu, Y. S. Ong, and M. Dash, "Wrapper-filter feature selection algorithm using a memetic framework," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 37, pp. 70-76, 2007.
- [4] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, pp. 1157-1182, 2003.

- [5] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80-98, 2015.
- [6] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary ant lion approaches for feature selection," *Neurocomputing*, 2016.
- [7] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014.
- [8] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371-381, 1/8/ 2016.
- [9] E. Emary, H. M. Zawbaa, C. Grosan, and A. E. Hassanien, "Feature subset selection approach by gray-wolf optimization," in *Afro-European Conference for Industrial Advancement*, 2015, pp. 1-13.
- [10] H. M. Zawbaa, E. Emary, B. PARV, and M. Shaarawi, "Feature Selection Approach based on Moth-Flame Optimization Algorithm," presented at the IEEE Congress on Evolutionary Computation, Vancouver, Canada, 2016.
- [11] S. Mirjalili, "Moth-flame optimization algorithm," *Know.-Based Syst.*, vol. 89, pp. 228-249, 2015.
- [12] J. H. Holland, *Adaptation in natural and artificial systems*: MIT Press, 1992.
- [13] J. Yang and V. G. Honavar, "Feature Subset Selection Using a Genetic Algorithm," *IEEE Intelligent Systems*, vol. 13, pp. 44-49, 1998.
- [14] O. Il-Seok, L. Jin-Seon, and M. Byung-Ro, "Hybrid genetic algorithms for feature selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1424-1437, 2004.
- [15] M. Mafarja and S. Abdullah, "Investigating memetic algorithm in solving rough set attribute reduction," *International Journal of Computer Applications in Technology*, vol. 48, pp. 195-202, 2013.
- [16] R. Jensen and Q. Shen, "Semantics-Preserving Dimensionality Reduction: Rough and Fuzzy-Rough-Based Approaches," *IEEE Trans. on Knowl. and Data Eng.*, vol. 16, pp. 1457-1471, 2004.
- [17] M. Anusha and J. Sathiseelan, "Feature Selection Using K-Means Genetic Algorithm for Multi-objective Optimization," *Procedia Computer Science*, vol. 57, pp. 1074-1080, 2015.
- [18] W. Jue, A. R. Hedar, and W. Shouyang, "Scatter Search for Rough Set Attribute Reduction," in *Bio-Inspired Computing: Theories and Applications, 2007. BIC-TA 2007. Second International Conference on*, 2007, pp. 236-240.
- [19] J. Kennedy and R. Eberhart, "Particle swarm optimization," presented at the Neural Networks, 1995. Proceedings., IEEE International Conference on, 1995.
- [20] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognition Letters*, vol. 28, pp. 459-471, 2007.
- [21] B. Chakraborty, "Feature subset selection by particle swarm optimization with fuzzy fitness function," in *Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on*, 2008, pp. 1038-1042.
- [22] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms," *Applied Soft Computing*, vol. 18, pp. 261-276, 2014.
- [23] R. Bello, Y. Gomez, A. Nowe, and M. M. Garcia, "Two-Step Particle Swarm Optimization to Solve the Feature Selection Problem," presented at the Proceedings of the Seventh International Conference on Intelligent Systems Design and Applications, Brazil, 2007.
- [24] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Erciyes University, Engineering Faculty, Computer Engineering Department, Kayseri/Türkiye2005.
- [25] J. Wang, T. Li, and R. Ren, "A real time idss based on artificial bee colony-support vector machine algorithm," presented at the Advanced Computational Intelligence (IWACI), 2010 Third International Workshop on, 2010.
- [26] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications*, vol. 27, pp. 1053-1073, 2016.
- [27] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH computer graphics*, vol. 21, pp. 25-34, 1987.
- [28] C. L. Blake and C. J. Merz. (1998, 1 June). *UCI Repository of machine learning databases*. Available: <http://www.ics.uci.edu/~mlearn/>